

**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ**

**CARRERA INFORMÁTICA**

**TESIS PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO  
EN INFORMÁTICA**

**TEMA:**

**NÚCLEO DE UN WORKFLOW DE PROCESOS  
ADMINISTRATIVOS EN LA ESPAM MFL**

**AUTORES:**

**DAYANA HELEN BAILÓN DELGADO  
JOSÉ SIMÓN MERA CANTOS**

**TUTOR:**

**ING. RAMÓN JOFFRE MOREIRA PICO, MG. SC**

**CALCETA, NOVIEMBRE 2017**

## **DERECHOS DE AUTORÍA**

Dayana Helen Bailón Delgado y José Simón Mera Cantos, declaran bajo juramento que el trabajo aquí descrito es de nuestra autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos los derechos de propiedad intelectual a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, según lo establecido por la Ley de Propiedad Intelectual y su reglamento.

.....  
**DAYANA H. BAILÓN DELGADO**

.....  
**JOSÉ S. MERA CANTOS**

## CERTIFICACIÓN DE TUTOR

Ramón Joffre Moreira Pico certifica haber tutelado la tesis NÚCLEO DE UN WORKFLOW DE PROCESOS ADMINISTRATIVOS EN LA ESPAM MFL, que ha sido desarrollada por Dayana Helen Bailón Delgado y José Simón Mera Cantos, previa la obtención del título de Ingeniero en Informática, de acuerdo al REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....  
**ING. RAMÓN J. MOREIRA PICO, MG. SC**

## APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del tribunal correspondiente, declaran que han APROBADO la tesis NÚCLEO DE UN WORKFLOW DE PROCESOS ADMINISTRATIVOS EN LA ESPAM MFL, que ha sido propuesta, desarrollada y sustentada por Dayana Helen Bailón Delgado y José Simón Mera Cantos, previa la obtención del título de Ingeniero en Informática, de acuerdo al REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....  
ING. MARLON R. NAVIA MENDOZA, MG.

**MIEMBRO**

.....  
ING. LIGIA E. ZAMBRANO SOLÓRZANO, MG

**MIEMBRO**

.....  
ING. DANIEL A. MERA MARTÍNEZ, MG.

**PRESIDENTE**

## **AGRADECIMIENTO**

A Dios por todas las bendiciones que diariamente nos brinda, permitiéndonos crecer personal y profesionalmente a través de las experiencias adquiridas a lo largo del incomparable camino de la vida;

A nuestros padres y hermanos por confiar en nosotros y apoyarnos continuamente en cada etapa de nuestro crecimiento profesional,

A la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López que nos dio la oportunidad de una educación superior de calidad y en la cual hemos forjado nuestros conocimientos profesionales día a día,

A aquellos docentes que supieron transmitirnos su afecto, compañerismo y conocimientos, los cuales han sido fundamentales para alcanzar esta meta y convertirnos en excelentes profesionales,

A nuestro tutor Ing. Joffre Moreira y al Ing. Javier López por habernos guiado durante el desarrollo de esta tesis y por ser nuestros amigos en el transcurso de nuestra vida universitaria,

A la Ing. Jéssica Morales, por habernos transmitido entusiasmo y motivación desde el inicio de nuestra carrera, siendo docente y amiga, y

A nuestros amigos con quienes hemos compartido grandes e inolvidables momentos, dentro y fuera de las aulas de clases.

**Los Autores**

## **DEDICATORIA**

A Dios, por permitirme ser quien soy.

A mis padres, por ser mi apoyo incondicional en cada momento de mi vida, por estar siempre ahí, dispuestos a ayudarme y sobre todo por confiar plenamente en que puedo crecer personal y profesionalmente cada día y dar lo mejor de mí;

A mis hermanos, porque siempre han confiado en mí, brindándome su cariño y las mejores alegrías diariamente.

**Dayana H. Bailón Delgado**

## **DEDICATORIA**

A Dios por darme salud permitiéndome alcanzar este momento tan especial en mi vida, por haberme dado la fortaleza necesaria en los momentos de dificultad y en los que he estado a punto de caer.

A mis padres Simón y Carmen por su apoyo incondicional a lo largo de mi vida, por darme la oportunidad de estudiar y ser una mejor persona cada día.

A mis hermanos por estar siempre pendientes del desarrollo de mi profesión y apoyarme en todo momento.

**José S. Mera Cantos**

## CONTENIDO GENERAL

CARÁTULA .....	i
DERECHOS DE AUTORÍA .....	ii
CERTIFICACIÓN DE TUTOR .....	iii
APROBACIÓN DEL TRIBUNAL.....	iv
AGRADECIMIENTO.....	v
DEDICATORIA.....	vi
DEDICATORIA.....	vii
CONTENIDO GENERAL.....	viii
CONTENIDO DE CUADROS Y FIGURAS.....	x
RESUMEN .....	xiv
PALABRAS CLAVES .....	xiv
ABSTRACT .....	xv
KEY WORDS .....	xv
CAPÍTULO I. ANTECEDENTES .....	1
1.1.  PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA .....	1
1.2.  JUSTIFICACIÓN .....	3
1.3.  OBJETIVOS .....	5
1.3.1.  OBJETIVO GENERAL .....	5
1.3.2.  OBJETIVOS ESPECÍFICOS .....	5
1.4.  IDEA A DEFENDER.....	6
CAPÍTULO II. MARCO TEÓRICO.....	7
2.1.  RESEÑA HISTÓRICA DE LA ESPAM MFL.....	7
2.1.1.  MISIÓN Y OBJETIVOS .....	7
2.2.  TECNOLOGÍAS ORIENTADAS AL PROCESO DE NEGOCIO .....	8
2.2.1.  SISTEMA DE GESTIÓN DE PROCESOS DE NEGOCIO (BPMS).....	8
2.2.2.  SISTEMA DE GESTIÓN DE FLUJOS DE TRABAJO (SGFT) .....	12
2.3.  TECNOLOGÍAS DE DESARROLLO.....	18
2.3.1.  VISUAL STUDIO.NET.....	18
2.3.2.  INTERNET INFORMATION SERVICES (IIS) .....	20
2.3.3.  SQL SERVER EXPRESS .....	21
2.3.4.  SQL SERVER AGENT .....	21
2.3.5.  SWAGGER .....	21

2.4.	METODOLOGÍA DE DESARROLLO .....	22
2.4.1.	METODOLOGÍA EVOLUTIVA INCREMENTAL PARA DESARROLLO DE APLICACIONES DE WORKFLOW (MEIDAW) .....	23
2.5.	INTEROPERATIVIDAD, API E INTERFACES DE SISTEMAS .....	34
2.5.1.	REST.....	34
2.5.2.	REST EN LA PRÁCTICA: HTTP.....	35
2.5.3.	URIS.....	36
2.5.4.	VERBOS HTTP.....	37
2.5.5.	TIPO MIME .....	38
2.5.6.	CÓDIGOS DE ESTADO.....	38
	CAPÍTULO III. DESARROLLO METODOLÓGICO .....	40
3.1.	MEIDAW .....	40
3.1.1.	FASE 1: ORIENTACIÓN AL PROCESO.....	41
3.1.2.	FASE 2: ORIENTACIÓN AL PRODUCTO .....	45
	CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	67
4.1.	RESULTADOS.....	67
4.1.1.	REALIZAR EL PLAN DEL PROYECTO, IDENTIFICANDO LOS REQUERIMIENTOS.....	67
4.1.2.	ELABORAR EL NÚCLEO CON BASE A LOS REQUERIMIENTOS ESTIPULADOS .....	81
4.1.3.	EJECUTAR PRUEBAS DE FUNCIONAMIENTO DEL NÚCLEO.....	97
4.1.4.	IMPLEMENTAR EL NÚCLEO EN LOS SERVIDORES DE LA INSTITUCIÓN .....	100
4.2.	DISCUSIÓN .....	104
	CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....	106
5.1.	CONCLUSIONES.....	106
5.2.	RECOMENDACIONES .....	107
	BIBLIOGRAFÍA .....	108

## CONTENIDO DE CUADROS Y FIGURAS

<b>Cuadro 2. 1.</b> Formato de una URL. ....	36
<b>Cuadro 2. 2.</b> Verbos HTTP más comunes. ....	38
<b>Cuadro 2. 3.</b> Códigos de respuestas más usados. ....	39
<b>Cuadro 3. 1.</b> Posibles estados de los elementos del flujo de proceso. ....	47
<b>Cuadro 3. 2.</b> Resumen y descripción de las plataformas y tecnologías de desarrollo. ....	52
<b>Cuadro 3. 3.</b> Identificación de actividades. ....	52
<b>Cuadro 3. 4.</b> Identificación de eventos. ....	54
<b>Cuadro 4. 1.</b> Usuarios interesados en el Plan de Proyecto. ....	67
<b>Cuadro 4. 2.</b> Resumen del Plan de Proyecto. ....	68
<b>Cuadro 4. 3.</b> Requerimientos generales del núcleo. ....	68
<b>Cuadro 4. 4.</b> Herramientas y lenguaje de programación para el desarrollo del núcleo. ....	69
<b>Cuadro 4. 5.</b> Lista de entregables. ....	70
<b>Cuadro 4. 6.</b> Cronograma de desarrollo del proyecto. ....	71
<b>Cuadro 4. 7.</b> Personal involucrado en el desarrollo del proyecto. ....	72
<b>Cuadro 4. 8.</b> Resumen del documento de Alcance y Requerimientos. ....	72
<b>Cuadro 4. 9.</b> Características de los usuarios. ....	73
<b>Cuadro 4. 10.</b> Restricciones, suposiciones y dependencias del núcleo. ....	73
<b>Cuadro 4. 11.</b> Conceptos de los elementos de los procesos administrativos. .	74
<b>Cuadro 4. 12.</b> Conceptos de otros elementos de los procesos administrativos. ....	74
<b>Cuadro 4. 13.</b> Resumen de los requisitos funcionales del núcleo. ....	75
<b>Cuadro 4. 14.</b> Requisitos comunes de las interfaces. ....	76
<b>Cuadro 4. 15.</b> Incremento 004. ....	78
<b>Cuadro 4. 16.</b> Incremento 011. ....	79
<b>Cuadro 4. 17.</b> Procedimientos almacenados del sistema agrupados por esquema. ....	87

<b>Cuadro 4. 18.</b> Procedimientos almacenados del sistema para notificaciones y reportes. ....	89
<b>Cuadro 4. 19.</b> Procedimientos almacenados del sistema como herramienta de control. ....	90
<b>Cuadro 4. 20.</b> Entidades que intervienen en cada requerimiento funcional. ....	92
<b>Cuadro 4. 21.</b> Controladores que intervienen en cada requerimiento funcional. ....	94
<b>Cuadro 4. 22.</b> Tiempos de ejecución de los recursos web. ....	99
<b>Figura 2. 1.</b> Ciclo de vida extendido de un proceso de negocio. ....	10
<b>Figura 2. 2.</b> Ciclo de vida de un proceso. ....	10
<b>Figura 2. 3.</b> Características de los BPMS. ....	11
<b>Figura 2. 4.</b> Estándares necesarios para implementar BPMS. ....	12
<b>Figura 2. 5.</b> Funcionamiento de los sistemas de workflow. ....	13
<b>Figura 2. 6.</b> Funcionamiento del núcleo durante la instancia de nuevo proceso. ....	13
<b>Figura 2. 7.</b> Beneficios tangibles e intangibles de los SGFT. ....	14
<b>Figura 2. 8.</b> Estándares de la WfMC. ....	15
<b>Figura 2. 9.</b> Modelo de referencia de la WfMC. ....	16
<b>Figura 2. 10.</b> Funcionamiento del Framework Entity ....	20
<b>Figura 2. 11.</b> Elementos que intervienen en la Orientación al proceso de MEIDAW. ....	24
<b>Figura 2. 12.</b> Esquema de la Orientación al proceso de MEIDAW. ....	27
<b>Figura 2. 13.</b> Esquema de la Orientación al Producto de MEIDAW. ....	28
<b>Figura 2. 14.</b> Ejemplo de petición HTTP. ....	36
<b>Figura 2. 15.</b> Ejemplo de respuesta HTTP. ....	36
<b>Figura 3. 1.</b> Esquema general de la gestión de los procesos administrativos. ....	41
<b>Figura 3. 2.</b> Documentos obtenidos en el Análisis del sistema. ....	45
<b>Figura 3. 3.</b> Estados y transición durante la ejecución de un proceso administrativo. ....	49
<b>Figura 3. 4.</b> Estados y transición de las tareas o trámites durante la ejecución de un proceso. ....	49

<b>Figura 3. 5.</b> Estados y transición de la comunicación (notificaciones y alertas) entre los usuarios, durante la ejecución del proceso.....	49
<b>Figura 3. 6.</b> Estados y transición de la ejecución de las tareas durante la creación o diseño del proceso.....	50
<b>Figura 3. 7.</b> Estados y transición de los mensajes enviados a los usuarios, durante el diseño del proceso. ....	50
<b>Figura 3. 8.</b> Código SQL del Sp_GuardarCabecera. ....	60
<b>Figura 3. 9.</b> Código C# del método GuardarCabeceraTarea. ....	60
<b>Figura 3. 10.</b> Código C# para el método POST para guardar una instancia. ..	61
<b>Figura 3. 11.</b> Vista de swagger en la ejecución de un método. ....	62
<b>Figura 3. 12.</b> Interfaz de la API, enlistando las entidades de los recursos disponibles. ....	95
<b>Figura 4. 1.</b> Esquema general del funcionamiento del núcleo. ....	69
<b>Figura 4. 2.</b> Diagrama UML para el RF01.....	77
<b>Figura 4. 3.</b> Diagrama UML del RF012.....	77
<b>Figura 4. 4.</b> Diagrama 1 - Gestión de los procesos administrativos, con sus respectivas versiones.....	81
<b>Figura 4. 5.</b> Diagrama 2 - Ejecución (instancia) de los procesos administrativos. ....	82
<b>Figura 4. 6.</b> Diagrama 3- Login de los usuarios administrativos de la ESPAM MFL, con sus roles y características. ....	83
<b>Figura 4. 7.</b> Diagrama 4 - Envío y recepción de notificaciones y mensajes para los usuarios. ....	84
<b>Figura 4. 8.</b> Diagrama 5 - Establecer las tareas y responsables para armar un nuevo proceso (o versión) administrativo.....	85
<b>Figura 4. 9.</b> Storage procedure para la tabla usuarios.....	86
<b>Figura 4. 10.</b> Diagrama de clases sobre la creación y ejecución de los procesos administrativos de la ESPAM MFL. ....	91
<b>Figura 4. 11.</b> Métodos HTTP disponibles para CabeceraTarea.....	95
<b>Figura 4. 12.</b> Detalles para ejecutar el método POST de la Cabecera. ....	96
<b>Figura 4. 13.</b> Código de respuesta para la URI api/cabecera. ....	96
<b>Figura 4. 14.</b> Ejemplos de pruebas de unidad aplicadas .....	97

**Figura 4. 15.** Ejemplo de prueba funcional aplicada..... 98  
**Figura 4. 16.** Esquema de ejecución del núcleo del workflow. .... 102

## **RESUMEN**

El objetivo general de este trabajo fue desarrollar el núcleo de un Workflow en la ESPAM MFL, para facilitar la automatización de sus procesos administrativos, para lo cual se utilizó MEIDAW. La primera fase de la metodología está orientada al proceso, donde se contempló el desarrollo de los documentos técnicos, mediante los cuales se identificaron: el alcance y requerimientos del sistema, el plan de desarrollo del núcleo, así como la planificación, ejecución y control de los incrementos. En la segunda fase se llevó a cabo el desarrollo del núcleo con base a los incrementos planificados en la fase anterior, mediante la ejecución de las siguientes actividades: análisis, diseño y desarrollo del sistema, ejecución de las pruebas unitarias, de regresión y funcionales, entrega de los documentos formales e instalación del sistema. Se obtuvo como resultado final una API web REST que sigue las recomendaciones de la WfMC en cuanto al desarrollo del núcleo de los SGFT, implementada en los servidores de la institución para el diseño, ejecución y supervisión de sus procesos administrativos, permitiendo que lleven un control de calidad de las actividades, ya sean manuales o automáticas, que componen los procedimientos automatizados mediante la herramienta.

## **PALABRAS CLAVES**

MEIDAW, automatización de procesos, gestor de procesos, BPMN 2.0, diagramas de proceso.

## **ABSTRACT**

The general objective of this work was to develop the core of a Workflow in ESPAM MFL, to facilitate the automation of its administrative processes, for which MEIDAW was used. The first phase of the methodology is oriented to the process, which included the development of technical documents, which identified: the scope and requirements of the system, the core development plan, as well as the planning, execution and control of The increments. In the second phase, the development of the kernel was carried out based on the increases planned in the previous phase, through the execution of the following activities: analysis, design and development of the system, execution of functional tests and data loading, Delivery of the formal documents and installation of the system. The final result was a REST web API that follows the recommendations of the WfMC regarding the development of the SGFT core, implemented in the institution's servers for the design, execution and supervision of its administrative processes, allowing them to control quality of the activities, whether manual or automatic, that compose the automated procedures using the tool.

## **KEY WORDS**

MEIDAW, process automation, process manager, BPMN 2.0, process diagrams.

# **CAPÍTULO I. ANTECEDENTES**

## **1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA**

Las Tecnologías de Información (TI) en las organizaciones, permiten optimizar el manejo e integración del tratamiento de los datos en todas sus etapas: entrada, procesamiento y salida, a través de sistemas web, servidores, redes de computadoras, entre otras herramientas tecnológicas, que facilitan la gestión del flujo de los procesos de cualquier área, incluyendo el ámbito administrativo (Aguilar, 2012).

Actualmente, los Sistemas de Gestión de Procesos de Negocio (SGPN) en contraposición a la falta de libertad y rigidez de las aplicaciones desarrolladas a la medida, mejoran los procesos; brindando flexibilidad de acuerdo a las necesidades y optimizando la circulación de información y el control de procesos, mediante la normalización de los métodos de trabajo, a través de una sola plataforma o núcleo principal que optimiza tales procesos y a su vez brinda información relevante para la toma de decisiones (Robledo, 2011).

Sin embargo, muchas instituciones no disponen de personal especializado en TI, equipos, materiales, software, recursos económicos y/o disponibilidad de tiempo para alcanzar dicha automatización, por lo cual no pueden acceder a la información de todos sus sistemas de manera simultánea y tienden a que sus aplicaciones pasen a ser obsoletas porque no pueden adaptarse a las necesidades y cambios del entorno.

La Unidad de Producción de Software (UPS) ubicada en la carrera de Computación de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López (ESPAM MFL) tiene como finalidad desarrollar aplicaciones académicas, pero es necesario que se desarrollen sistemas para el área administrativa, mismos que puedan adaptarse a los cambios de los procesos o

necesidades de los diferentes departamentos, y que además se integren en un solo sistema, para evitar inconvenientes al momento de realizar el intercambio de información.

Por ello, la ESPAM MFL requiere automatizar sus procesos administrativos con el fin de gestionar los documentos, información o tareas de forma eficiente entre un participante, sistema o departamento, y otro, respetando el cumplimiento de lo establecido en el reglamento de procesos. Por lo cual, los autores se plantearon la siguiente interrogante:

¿Cómo facilitar la automatización de los procesos administrativos de la ESPAM MFL?

## 1.2. JUSTIFICACIÓN

De acuerdo al objetivo 1 del Plan Nacional del Buen Vivir en su sección Políticas y Lineamientos Estratégicos, literal a del punto 1.5, establece: “Agilizar y simplificar los procesos y procedimientos administrativos, con el uso y el desarrollo de tecnologías de información y comunicación”, brindando una mejor atención a la ciudadanía (SENPLADES, 2013). Además, según lo dispuesto en el art. 6 de la Norma Técnica de Administración por Procesos, en su literal b, se indica que la automatización de procesos es: “un grupo de actividades mediante las cuales se lleva el flujo de un proceso manual hacia una herramienta de software, que representan sus entradas, salidas o entregables, su recorrido, roles, usuarios y datos necesarios para cada actividad del proceso”, lo que permite a la institución llevar un control en lo referente a tiempos de ejecución, cronología, camino recorrido, actividades complementarias y participantes (Registro Oficial, 2013).

A fin de contribuir a tales disposiciones legales, a través del desarrollo del núcleo de una plataforma de generación de aplicaciones de flujo de trabajo, los autores plantearon facilitar la automatización de los procesos administrativos de la ESPAM MFL, aportando con la gestión de sus procesos de negocio, misma que es considerada por la Workflow Management Coalition (2014) como una disciplina que involucra cualquier combinación de modelado, automatización, ejecución, control, medición y optimización de los flujos de actividad empresarial, en apoyo de los objetivos de la empresa, que abarca sistemas, empleados, clientes y socios dentro y fuera de los límites de la empresa.

Por lo cual, el núcleo del Workflow mejorará significativamente la gestión administrativa de la ESPAM MFL, ya que gestionará el flujo de los procesos administrativos implementados en dicha herramienta, y a su vez, facilitará la reingeniería de los mismos. Además, permitirá ahorrar recursos económicos, ya que tales procesos podrán ser ejecutados a tiempo, la información tratada podrá ser consultada en cualquier momento y mediante integración de datos,

se podrá conectar a los sistemas existentes de dicha institución, contribuyendo a la centralización de información.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO GENERAL**

Desarrollar el núcleo de un Workflow en la ESPAM MFL, para facilitar la automatización de sus procesos administrativos.

#### **1.3.2. OBJETIVOS ESPECÍFICOS**

- Realizar el plan del proyecto, identificando los requerimientos.
- Elaborar el núcleo con base a los requerimientos estipulados.
- Ejecutar pruebas de funcionamiento del núcleo.
- Implementar el núcleo en los servidores de la Institución.

#### **1.4. IDEA A DEFENDER**

El desarrollo del núcleo de un Workflow facilitará la automatización de los procesos administrativos de la ESPAM MFL.

## **CAPÍTULO II. MARCO TEÓRICO**

### **2.1. RESEÑA HISTÓRICA DE LA ESPAM MFL**

La información detallada a continuación ha sido publicada por la ESPAM-MFL (s.f.) en su sitio web oficial, estableciendo que Calceta es una ciudad rica en recursos agrícolas y pecuarios, por lo que fue necesario contar con un centro de estudios superiores que manejaran estas áreas de manera que la población estudiantil, con dificultad para trasladarse a universidades fuera de la zona, pudiera alcanzar un título académico.

Además, detalla que la ESPAM inicia sus labores con las carreras de Agroindustria, Medio Ambiente, Agrícola y Pecuaria. Posteriormente, se crea la carrera de Informática, Administración de Empresas Pública y Privada, y finalmente Turismo. Desde su creación en el año 1999, hasta la actualidad, se han establecido procesos que han permitido la gestión administrativa de la institución.

#### **2.1.1. MISIÓN Y OBJETIVOS**

Esta tesis aporta a la filosofía establecida por la ESPAM MFL, que tiene como misión la formación integral y continua de profesionales que contribuyan de forma proactiva y creativa al desarrollo cultural, económico, político y social sostenible de su entorno y nación, según lo publicado por Derecho Ecuador (2012). Igualmente, dicha fuente establece los objetivos estratégicos de la institución, y este trabajo contribuye al literal B: formación en las diversas especialidades académicas, equipos profesionales en los campos humanísticos, científicos y tecnológicos, mediante la ejecución de la tesis de grado correspondiente al desarrollo de una herramienta tecnológica que automatiza la ejecución de sus procesos administrativos.

## **2.2. TECNOLOGÍAS ORIENTADAS AL PROCESO DE NEGOCIO**

Flores *et al.* (2014) destacan que el desempeño eficiente de una organización depende del nivel de eficiencia de sus procesos. Tales autores citan a Pérez (2007), quien define un proceso como una secuencia ordenada de actividades repetitivas, transformando una entrada en una salida con valor añadido para el usuario.

Puntualizando esta definición general, a nivel de empresa u organización, según Hitpass (2014) citado por Calle *et al.* (2016) se maneja el concepto de proceso de negocio, que no es más que la forma como interaccionan los recursos (humanos, materiales) y actividades a través de las distintas áreas funcionales de la organización, con el fin de generar productos u ofertar servicios.

Por lo cual, Delgado y Calegari (2014) enfatizan que la idea de las entidades es eliminar la brecha existente entre las áreas que definen los objetivos de la empresa y la de Tecnologías de la Información (TI) independientes y según Flores *et al.* (2014) se puede realizar mediante la implementación de tecnologías que den soporte a su Gestión de Procesos de Negocio (BPM, Business Process Management), lo que se conoce como BPMS (BPM System), permitiendo la automatización de los procesos, su rediseño y mejora continua.

### **2.2.1. SISTEMA DE GESTIÓN DE PROCESOS DE NEGOCIO (BPMS)**

De acuerdo a lo especificado por Garimella *et al.* (2008) citados por Flores *et al.* (2014) y Karabogolian *et al.* (2014), es un conjunto completo de software que facilita todos los aspectos de la BPM, como diseño de procesos, flujo de trabajo, aplicaciones, integración y supervisión de la actividad, para entornos centrados tanto en los sistemas como en el ser humano. Es decir, los BPMS integran conceptos del área de administración de Negocios y las TI.

### **2.2.1.1. GESTIÓN DE PROCESOS DE NEGOCIO (BPM)**

De acuerdo a CBOK (2009) citado por Flores *et al.* (2014), es una disciplina enfocada a diseñar, ejecutar, documentar, medir, monitorear y controlar procesos de negocio tanto automatizados como no automatizados, con el fin de obtener resultados alineados a los objetivos estratégicos de la organización.

Teniendo como propósito fundamental, según Llanes-Font *et al.* (2014) mejorar la eficiencia y eficacia de la organización para alcanzar los objetivos definidos, contribuyendo significativamente a la entidad al permitir disponer de la información requerida en el momento oportuno, de acuerdo a lo indicado por Artilles y Márquez (2013).

#### **2.2.1.1.1. IMPORTANCIA DE BPM**

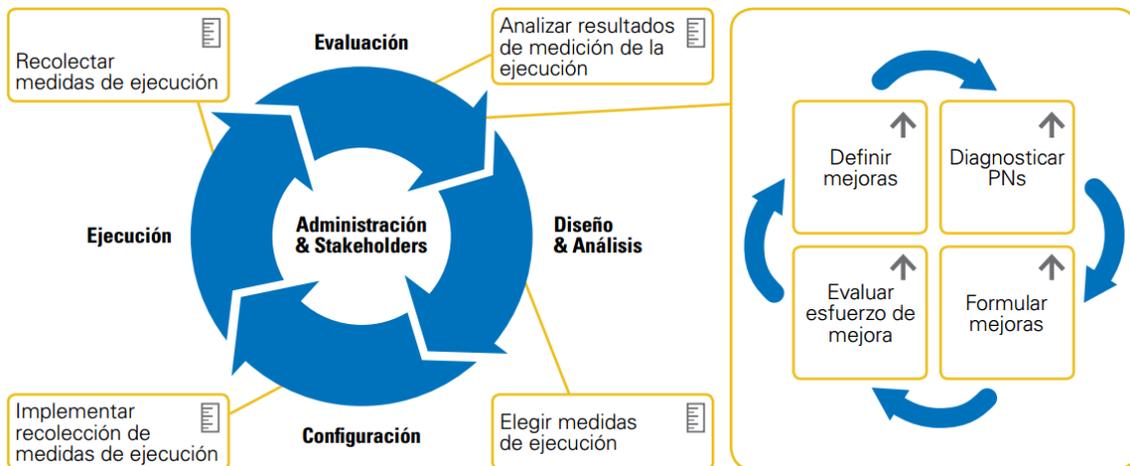
Robledo (2011) enfatiza que la implantación de la BPM se ha revelado como la herramienta de gestión empresarial más efectiva para todo tipo de organización. Reflexionando en este sentido, Karabogolian *et al.* (2014) especifican que al ser BPM un paradigma naturalmente integrador, ha cobrado importancia dentro del área de tecnología informática, la cual en los últimos años ha evolucionado desde el concepto de producto hacia el paradigma de servicios y soluciones.

Es por esto, que además del desarrollo de este documento, tal como lo mencionan Karabogolian *et al.* (2014), BPM y sus herramientas de soporte están siendo incorporados por muchos estudiantes como líneas de investigación, para desarrollar su tesina e interactuar con docentes e investigadores formados, a fin de brindar soluciones a problemas reales.

#### **2.2.1.1.2. CICLO DE VIDA BPM**

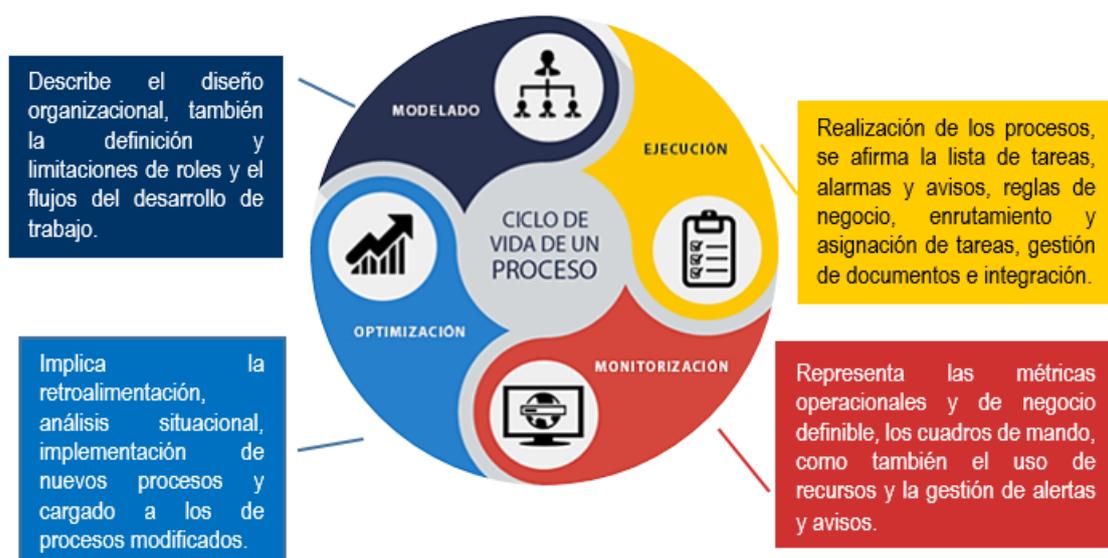
Weske (2007), mencionado por Delgado y Calegari (2014), detalla que el ciclo de vida consiste básicamente en cuatro fases: Análisis y Diseño, Configuración, Ejecución y Evaluación. Por su parte, Delgado (2011) también citado por dichos

autores, explica que puede ser extendido para incorporar actividades específicas para medición y mejora continua, tal como lo muestra la Figura 2. 1.



**Figura 2. 1.** Ciclo de vida extendido de un proceso de negocio.  
Fuente: Delgado (2011) y Weske (2007) citados por Delgado y Calegari (2014).

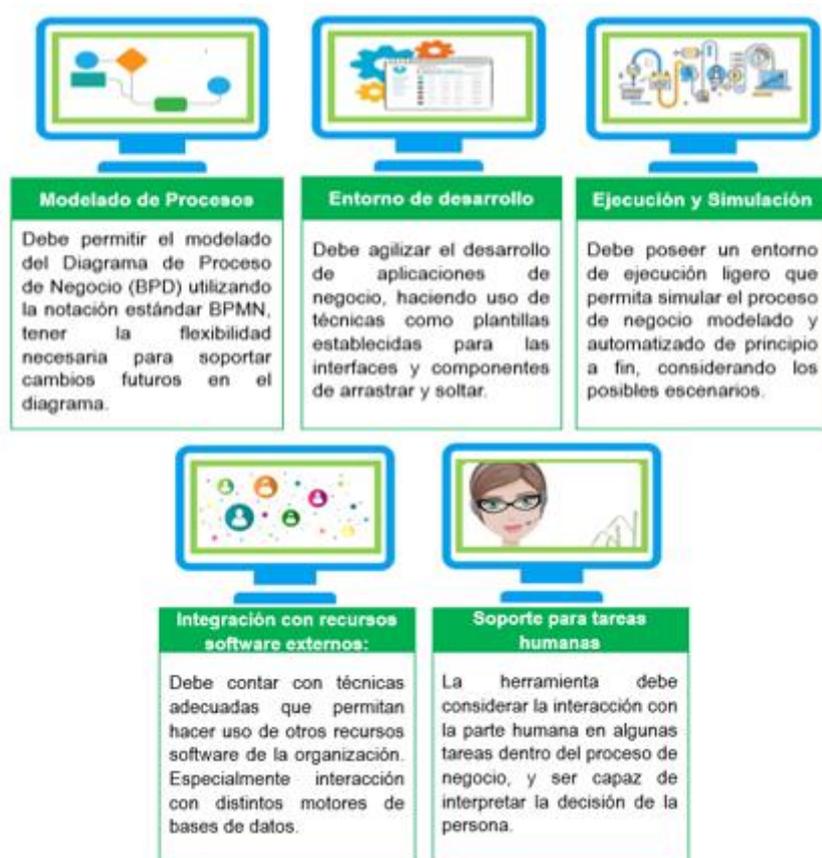
Con base a lo expuesto anteriormente, Carrasco (2006) citado por Cruz y Lévano (2013) resumen las cuatro fases mencionadas y las respectivas actividades de mejoras y optimización, en las siguientes etapas: Modelado, Ejecución, Monitorización y Optimización, definiendo así el ciclo de vida de un proceso, como se presenta en la Figura 2. 2.



**Figura 2. 2.** Ciclo de vida de un proceso.  
Fuente: Carrasco (2006) citado por Cruz y Lévano (2013).

### 2.2.1.2. CARACTERÍSTICAS DE LOS BPMS

Bazán *et al.* (2010) y Web Ratio (2012) citados por Flores *et al.* (2014) han especificado un conjunto de características (Figura 2. 3) que debe tener un BPMS para poder responder a las exigencias del problema. Destacando que pueden variar dependiendo del tipo de problema que se espera satisfacer.

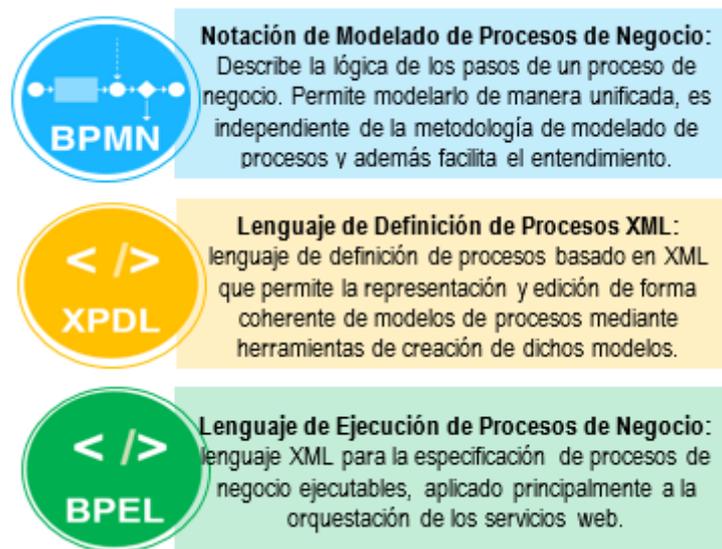


**Figura 2. 3.** Características de los BPMS

**Fuente:** Autores, en base a la información de Bazán *et al.* (2010) y Web Ratio (2012).

### 2.2.1.3. ESTÁNDARES EN BPMS

Cuando se implementan los Procesos de Negocio de la entidad al BPMS, se pueden emplear los estándares presentados en la Figura 2. 4. Aunque no es necesario que se apliquen todos, algunos autores como Garimella *et al.* (2008) y WfMC (2012) citados por Flores *et al.* (2014), recomiendan su uso.



**Figura 2. 4.** Estándares necesarios para implementar BPMS.

**Fuente:** Autores, en base a la información de Garimella *et al.* (2008) y WfMC (2012).

## 2.2.2. SISTEMA DE GESTIÓN DE FLUJOS DE TRABAJO (SGFT)

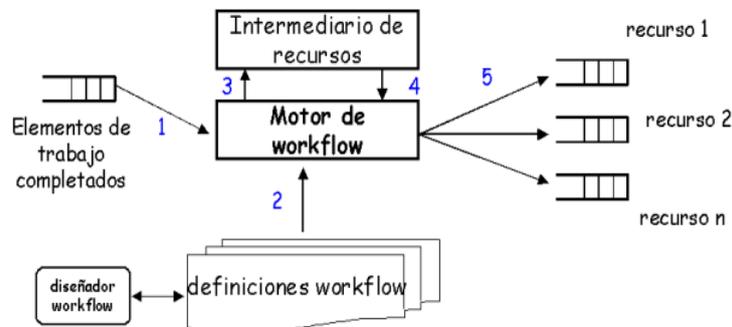
Marrero (2015) señalan que, en la antigüedad, cualquier tarea perteneciente a un proceso, exigía la presencia de un usuario humano. Por lo cual, Menéndez y Castellanos (2016) indican que se originaron los SGFT como una forma de reducir el tiempo y costo al llevar a cabo los procesos de negocios y asegurar la calidad de la ejecución de tareas.

Profundizando en este tema, Hollingsworth (1995) citado por Menéndez y Castellanos (2016) detalla que un SGFT (también llamado simplemente Workflow) brinda una automatización de los procesos de negocios mediante la gestión de la secuencia de actividades que los componen, asignándolas a personas o procesos informáticos que las llevarán a cabo. Además, Peralta *et al.* (2013) destaca que es una de las metodologías más significativas para el soporte de dicha automatización.

### 2.2.2.1. FUNCIONAMIENTO

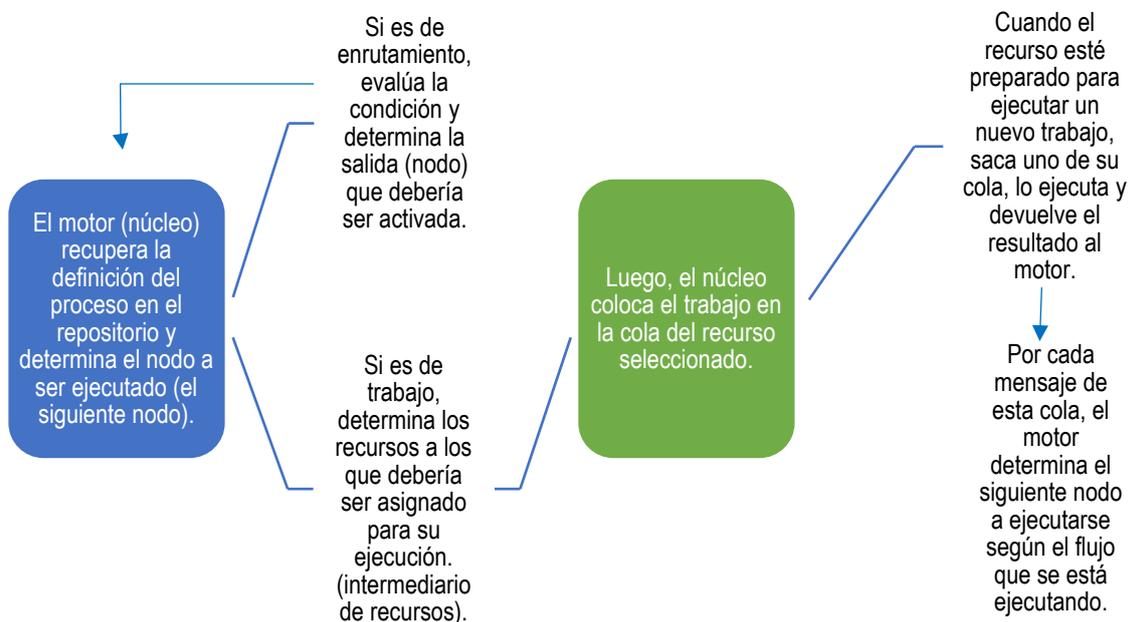
De Soto y Fernández (2006) explican el funcionamiento de los SGFT en la Figura 2. 5, indicando que las instancias son ejecutadas por un motor de workflow. Este motor es básicamente un planificador que organiza el trabajo a

realizar y lo asigna al actor que se encargará de realizarlo, que en la terminología de workflow se denomina recurso.



**Figura 2. 5.** Funcionamiento de los sistemas de workflow.  
Fuente: De Soto y Fernández (2006).

Ellos explican la instancia de un nuevo proceso en los pasos y condiciones presentados en la Figura 2. 6.



**Figura 2. 6.** Funcionamiento del núcleo durante la instancia de nuevo proceso.  
Fuente: De Soto y Fernández (2006).

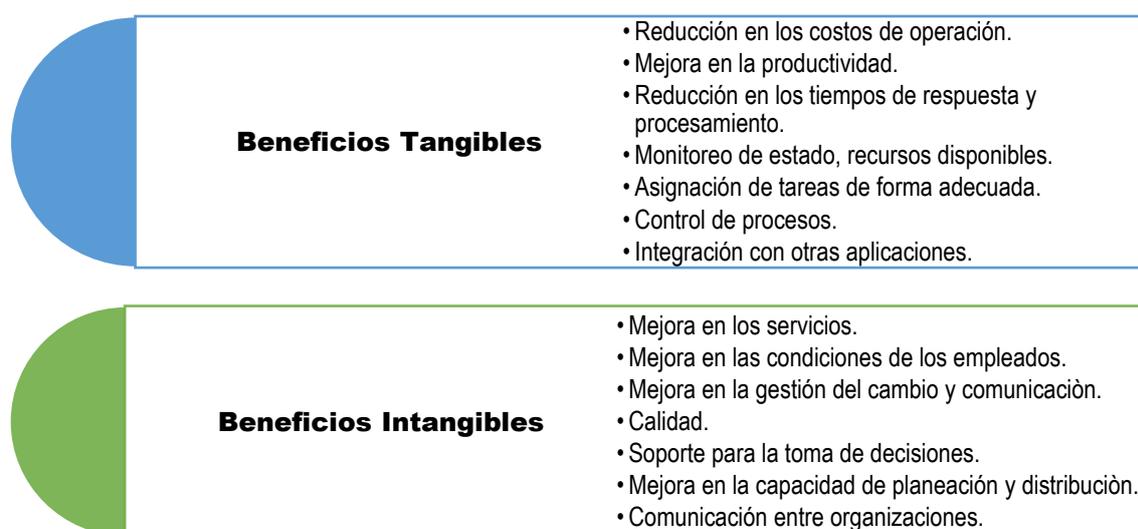
Reflexionando en esta explicación, de acuerdo a Plesums (2002) citado por Menéndez y Castellanos (2016) las funciones más comunes que proporcionan los SGFT son:

- Asignación de tareas al personal y notificación de las mismas.
- Permitir la colaboración en las tareas comunes.
- Optimización de recursos humanos y técnicos.
- Automatización de las secuencias de los procesos de negocio y optimización de las mismas.
- Agilización de los procesos de negocio.
- Control y seguimiento de dichos procesos.
- Interfaz para el acceso a sistemas de datos.
- Registro y monitoreo de acciones realizadas.

Por su parte, Menéndez y Castellanos (2016) destacan que la característica básica que debe proporcionar un sistema Workflow es la flexibilidad, a fin de que pueda adaptarse a los diferentes cambios que puedan surgir en el proceso de negocio y la organización en general.

#### 2.2.2.2. VENTAJAS

De acuerdo a Plesums (2002) y Prior (2003) citados por Menéndez y Castellanos (2016), los beneficios que se obtienen al emplear un SGFT pueden ser divididos en tangibles e intangibles y son presentados en la Figura 2. 7.

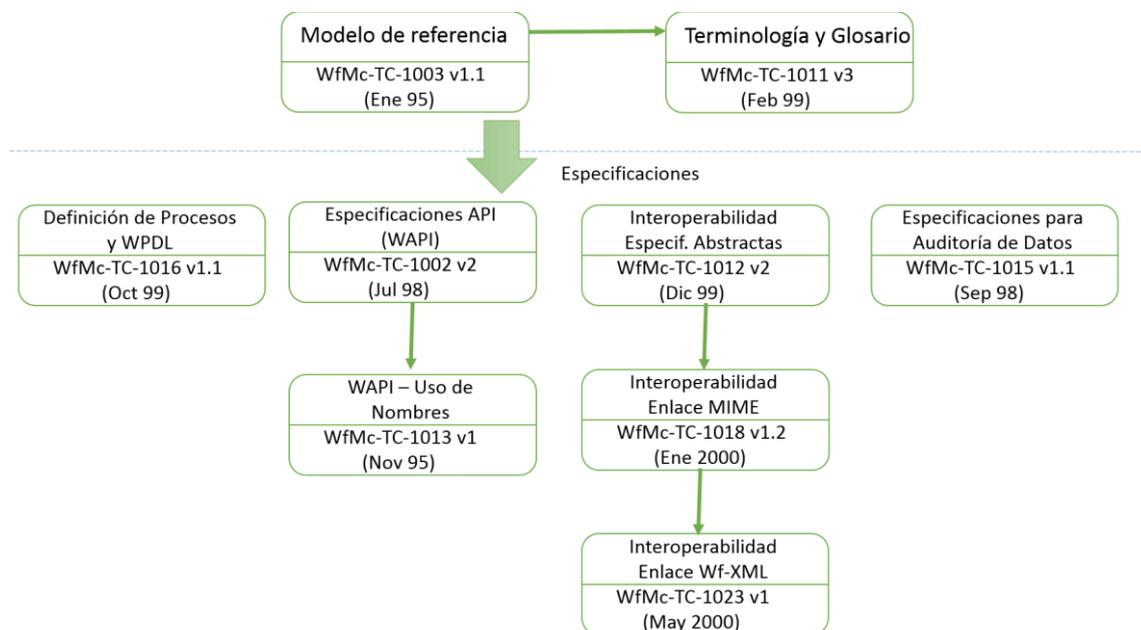


**Figura 2. 7.** Beneficios tangibles e intangibles de los SGFT.

Fuente: Plesums (2002) y Prior (2003) citados por Menéndez y Castellanos (2016).

### 2.2.2.3. ESTANDARIZACIÓN

Según lo indica Rodríguez (2011) la Workflow Management Coalition (WFMC) es una organización internacional, sin ánimo de lucro, que tiene como objetivo principal promover el uso de SGFT mediante el establecimiento de estándares que faciliten la creación, desarrollo y análisis de estos sistemas. Tales estándares son resumidos por la misma organización y se muestran en la Figura 2. 8.

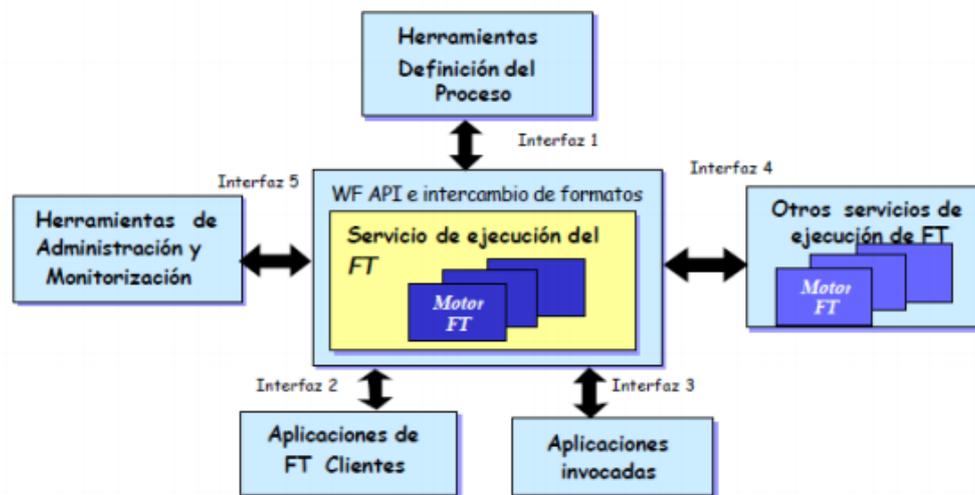


**Figura 2. 8.** Estándares de la WfMC  
Fuente: Rodríguez (2011).

En este sentido, Fresno (2010) detalla que la necesidad de estandarizar la forma de comunicación entre los distintos componentes de un Workflow nace con el fin de poder tener flexibilidad a la hora de operar con distintos productos. Esta necesidad se justifica por las proyecciones que se tienen actualmente sobre la penetración de la tecnología Workflow en el mercado en los próximos años.

#### 2.2.2.4. MODELO DE REFERENCIA

Con el propósito de establecer un estándar para el desarrollo de Sistemas Gestores de Flujos de Trabajo, la WfMC ha establecido un modelo de referencia para Sistemas Gestores de Flujos de Trabajo, según lo indicado por Hollingsworth (1995) citado por Menéndez y Castellanos (2016). Dicho modelo es mostrado en la Figura 2. 9.



**Figura 2. 9.** Modelo de referencia de la WfMC.  
Fuente: Hollingsworth (1995) citado por Menéndez y Castellanos (2016).

En el modelo adoptado existe una separación entre los procesos y el control de la lógica de las actividades, la cual permite la integración de las diversas herramientas con una aplicación particular. A continuación, se detallan los elementos que intervienen en dicho modelo.

##### 2.2.2.4.1. MOTOR DE WORKFLOW

De acuerdo a Hollingsworth (1995) citado por Menéndez y Castellanos (2016) y lo especificado por Zorzán *et al.* (2014), es el software que provee el control del ambiente de ejecución de una instancia de flujo de trabajo.

Profundizando en este tema, también indican que típicamente el motor provee facilidades para: interpretación de la definición de procesos; control de las

instancias de los procesos: creación, activación, terminación, etc.; navegación entre actividades; soporte de interacción con el usuario: control de datos al usuario o hacia aplicaciones; invocación de aplicaciones externas.

#### **2.2.2.4.2. SERVICIO DE EJECUCIÓN DE WORKFLOW**

Fresno (2010) y Hollingsworth (1995) citado por Menéndez y Castellanos (2016), indican que este componente interpreta la descripción de procesos y controla las diferentes instancias de los procesos, secuencia de actividades, adiciona elementos a la lista de trabajo de los usuarios, e invoca aplicaciones necesarias. Todas estas tareas son hechas por uno o más motores de Workflow, los cuales manejan la ejecución de las distintas instancias de varios procesos.

#### **2.2.2.4.3. INTERFAZ DE PROGRAMACIÓN DE APLICACIONES DE WORKFLOW (WAPI)**

Según Fresno (2010) y Hollingsworth (1995) citado por Menéndez y Castellanos (2016), las WAPI pueden ser vistas como un conjunto de API's (Application Programming Interface) y funciones de intercambio soportadas por el servicio de representación de Workflow. Las WAPI permiten la interacción del servicio de representación de Workflow con otros recursos y aplicaciones.

- **Herramientas de definición de procesos (interfaz 1):** La salida del modelado y diseño de proceso es una “definición de procesos”, la cual puede ser interpretada en tiempo de ejecución por el o los motores de Workflow.
- **Aplicaciones clientes (interfaz 2):** La interfaz entre una aplicación cliente de Workflow y el motor de Workflow debe ser lo suficientemente flexible en los siguientes puntos: identificadores de procesos y actividades, estructuras de datos, diferentes alternativas de comunicación.
- **Aplicaciones invocadas (interfaz 3):** Está orientada a interactuar con agentes de una aplicación, o con toda la aplicación. Dichas aplicaciones

deben estar orientadas al contexto general de un sistema de Workflow, es decir, deben poder interactuar directamente con el motor de Workflow.

- **Funciones de interoperabilidad WAPI (interfaz 4):** Existen dos aspectos necesarios para la interoperabilidad: alcance de la interpretación común de la definición de procesos que será realizada; soporte en tiempo de ejecución para el intercambio de diferentes tipos de información de control y transferencia de los datos relevantes del Workflow, y/o de las aplicaciones entre los distintos servicios de representación.
- **Herramientas de administración y monitoreo (interfaz 5):** El propósito de esta interfaz es permitir una vista completa del estado del flujo de trabajo, además de poder realizar auditorías sobre los datos del sistema

### **2.3. TECNOLOGÍAS DE DESARROLLO**

Tal como lo expresa Arbeláez *et al* (2011) al momento de realizar un desarrollo de software, se recomienda seleccionar un modelo de acuerdo a la naturaleza del proyecto y de la aplicación. Por lo cual, se seleccionaron las siguientes herramientas, mismas que permitieron el desarrollo óptimo del núcleo o motor del Workflow.

#### **2.3.1. VISUAL STUDIO.NET**

De acuerdo a Arbeláez *et al.* (2011), Visual Studio.Net soporta varios lenguajes de programación como lo son Visual C#, Visual J#, C++, ASP.NET y Visual Basic.NET. Actualmente se han desarrollado las extensiones necesarias para muchos otros lenguajes de programación. En este aspecto, Microsoft (2011) citado por Arbeláez *et al.* (2011) detalla que Visual Studio es un Entorno de Desarrollo Integrado (IDE) que permite a los desarrolladores de software crear aplicaciones de escritorio, aplicaciones de tipo cliente servidor, aplicaciones y servicios web, entre otras.

### **2.3.1.1. VISUAL C SHARP (C#)**

Stein & Geyer-Schulz (2013) y Ceballos (2012) indican que es un lenguaje de programación que se destaca por su robustez y sencillez, basado en el paradigma POO y que combina el tipado de sistema estático junto con el dinámico, utilizando como estrategia de ejecución JIT-Compiled, mediante su Common Language Runtime (CLR), destinado para compilar Common Intermediate Language (CIL) al código de maquina nativo.

Según lo indican Feliciano *et al.* (2012), C# es un lenguaje de programación de los más nuevos, diseñado para crear un amplio número de aplicaciones empresariales que se ejecutan en .NET Framework. Supone una evolución de Microsoft C y Microsoft C++; es sencillo, moderno, proporciona seguridad de tipos y está orientado a objetos. Especifican que el código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de CLR. Estos servicios incluyen interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad y mayor compatibilidad entre versiones.

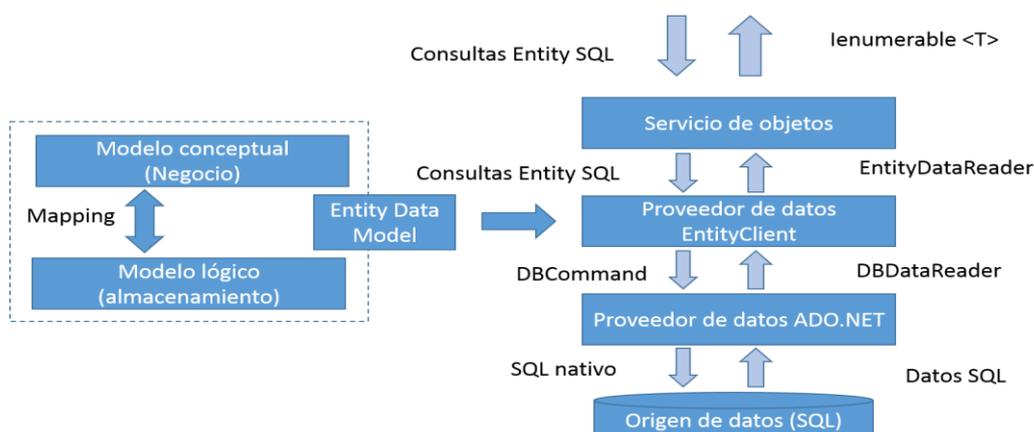
Por lo cual, los mencionados autores indican que C# se presenta como Visual C# en el conjunto de programas Visual Studio .NET, utilizando plantillas de proyecto, diseñadores, páginas de propiedades, asistentes de código, un modelo de objetos y otras características del entorno de desarrollo.

### **2.3.1.2. ASP .NET WEB API**

De acuerdo al sitio oficial de Microsoft (2015), ASP.NET Web API es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles. ASP.NET Web API es la plataforma perfecta para crear aplicaciones RESTful en .NET Framework.

### 2.3.1.3. ENTITY FRAMEWORK

De acuerdo con Brice-Arnaud (2016) la vocación del Framework Entity es generalizar el enfoque LINQ (Language Integrated Query) para conferirle dos nuevas características. Independencia de cara a SQL Server y la manipulación de verdaderos objetos gráficos de objetos de negocio, posibilitando definir relaciones de herencia entre ellos. La arquitectura de Entity aumenta progresivamente la abstracción para desarrollar esta doble independencia, las capas inferiores están destinadas a resolver la técnica de almacenamiento y las superiores a ofrecer interfaces con los procesamientos de negocios (Figura 2. 10)



**Figura 2. 10.** Funcionamiento del Framework Entity  
Fuente: Brice-Arnaud (2016)

### 2.3.2. INTERNET INFORMATION SERVICES (IIS)

De acuerdo con Morales *et al.* (2012) y Lazotmh (2010) citado por Avellán y Zambrano (2015) quienes coinciden al mencionar que Microsoft Internet Information Services (IIS) ofrece una plataforma de servidor web para el desarrollo, implementación, hospedaje y administración de sitios web, que tiene la capacidad de procesar distintos tipos de páginas desarrolladas en los lenguajes más populares utilizados en la Web, desde ASP.NET a PHP.

Morales *et al.*, (2012) mencionan además que el servidor IIS puede ser instalado en los sistemas operativos de Microsoft destinados a ofrecer servicios

Windows Server, o en sistemas operativos tales como Windows XP, Vista o 7, pero con algunas limitantes importantes.

### **2.3.3. SQL SERVER EXPRESS**

Martín *et al.* (2016) especifican que es un Sistema Gestor de Base de Datos (SGBD) basado en el lenguaje Transact-SQL, que según Fotache & Strimbei (2015) es capaz de analizar grandes conjuntos de tablas relacionales y el procesamiento de modelos estadísticos. Express se refiere a la versión liviana y gratuita.

Al profundizar sobre SGBD, Martín *et al.* (2016) también indican que permiten crear, modificar y mantener una Base de Datos, siendo capaz de poner a disposición de muchos usuarios grandes cantidades de datos, de manera simultánea.

### **2.3.4. SQL SERVER AGENT**

De acuerdo con el sitio oficial de Microsoft (2017), el Agente de SQL Server es un servicio de Microsoft Windows que ejecuta tareas administrativas programadas, denominadas trabajos (Jobs). Dichos trabajos pueden ser ejecutados según varios factores: programación, como respuesta a un evento específico o a petición. Por ejemplo, si desea realizar una copia de seguridad de todos los servidores de la organización todos los días entre semana después del horario de trabajo, puede automatizar esta tarea.

### **2.3.5. SWAGGER**

Dietz (2016) declara que Swagger (<http://swagger.io>) es una especificación y un framework ideal para crear la documentación interactiva de una API REST. Mientras que Varanasi & Belida (2015) contribuyen a dicha afirmación, explicando que el marco de Swagger se ha convertido en una gran comunidad con una variedad de herramientas construidas alrededor de su capacidad de proporcionar definiciones ricas de API, desde la generación de SDK para

generar código de cliente de API y la codificación de servidor a la documentación en vivo y los entornos de desarrollo API, permitiendo que la documentación esté sincronizada con cualquier cambio realizado en los servicios REST.

Además, Varanasi & Belida (2015) también detallan que Swagger es una especificación agnóstica del lenguaje con implementaciones disponibles para una variedad de lenguajes como Java, Scala y PHP. La especificación está compuesta por dos tipos de archivos: un archivo de lista de recursos y un conjunto de archivos de declaración de API que describen la API REST y las operaciones disponibles.

#### **2.4. METODOLOGÍA DE DESARROLLO**

Pons *et al.* (2010) citado por Marciszack *et al.* (2015) han expuesto que el problema de construir software debe ser encarado de la misma forma en que los ingenieros construyen otros sistemas complejos, como puentes, edificios, barcos y aviones. Por lo cual, indican que la idea básica consiste en observar el sistema de software a construir como un producto complejo y a su proceso de construcción como un trabajo ingenieril. Es decir, un proceso planificado basado en metodologías formales apoyadas por el uso de herramientas.

Al reflexionar en este sentido, Jiménez y Orantes (2012) puntualizan que al utilizar una metodología para desarrollo de software, se espera que pueda proveer un conjunto de prácticas y herramientas que faciliten el proceso de desarrollo, ofreciendo un producto con alta calidad, seguro y que satisfaga las expectativas del cliente. Sin embargo, las metodologías ágiles y tradicionales no son lo suficientemente sistémicas para adaptarse a los desarrollos de Workflow deseados. Es por esto que dichos autores también expresan que existen las metodologías híbridas, las cuales retoman las ventajas de las mencionadas, siendo una combinación de las mejores prácticas existentes dentro de ellas.

#### **2.4.1. METODOLOGÍA EVOLUTIVA INCREMENTAL PARA DESARROLLO DE APLICACIONES DE WORKFLOW (MEIDAW)**

De acuerdo a lo referido por Flores y Rufino (2007) durante la implementación de un sistema, es una metodología híbrida propuesta por Mendoza y Reynoso (2001) para desarrollar los distintos tipos de sistemas de Workflow, entre ellos los encargados de gestionar procesos administrativos. Fusiona elementos de la Metodología Evolutiva Incremental (MEI), como esquema de trabajo (enfocada al proceso), y el soporte de los estándares metodológicos de Workflow Management Coalition (WfMC), para obtener la calidad del producto.

Cabe destacar que hay actividades que son incrementadas durante dicha implementación realizada por Flores y Rufino (2007), algunas son referidas y utilizadas durante el desarrollo de este trabajo.

##### **2.4.1.1. LA ORIENTACIÓN AL PROCESO**

Existen elementos de las MEI que son consideradas por MEIDAW para el control de los proyectos y comprenden la definición de esta fase, la cual especifica cómo es el proceso de desarrollo. Tales elementos son explicados por Mendoza y Reynoso (2001) mencionados por Flores y Rufino (2007); en primer lugar se elaboran el Plan de proyecto y el documento de Alcance y Requerimientos y durante el desarrollo del proyecto, cada 15 días, se planifican los incrementos para posteriormente ejecutarse, luego se realiza el control de dichos incrementos y se presentan los respectivos Informes de avances (Figura 2. 11).

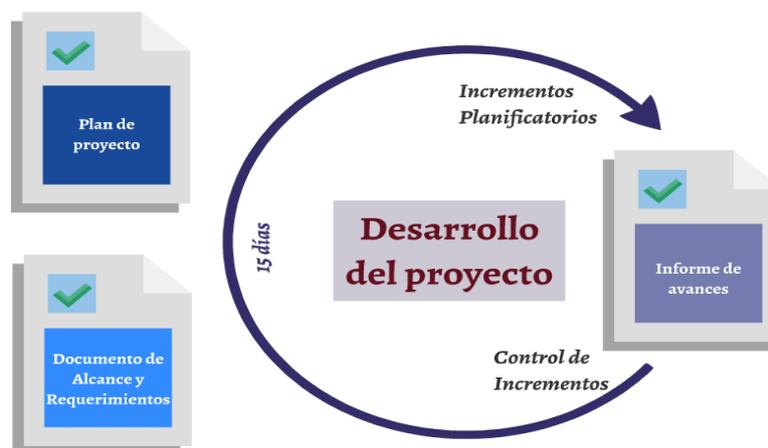


Figura 2. 11. Elementos que intervienen en la Orientación al proceso de MEIDAW.

Fuente: Autores

- **INCREMENTOS PLANIFICATORIOS**

MEIDAW recomienda que los incrementos planificatorios tengan una duración de quince (15) días. Sigue el esquema general de las metodologías tradicionales de desarrollo, que de acuerdo a Cervantes y Gómez (2012) se lo conoce como ciclo de vida del producto: análisis, diseño, implementación, pruebas, documentación e implantación. Mendoza y Reynoso (2001) citados por Flores y Rufino (2007) hacen énfasis a que estas etapas se encuentran levemente solapadas, es decir, las partes finales de una etapa se realizan en paralelo con el comienzo de la próxima, para ajustarse de mejor manera a la realidad y permitir la adaptabilidad a cambios futuros, reduciendo además el tiempo de finalización, si se aumentan los recursos.

- **PLAN DEL PROYECTO**

Salazar y Prado-Calderón (2013) consideran que la planificación es una de las etapas más significativas en toda investigación, y al reflexionar en esta consideración, Mendoza y Reynoso (2001) citados por Flores y Rufino (2007) han calificado el Plan de proyecto como el primer aspecto importante dentro de la metodología, el cual permite realizar una macro-planificación del proyecto, abarcando las siguientes especificaciones:

- Identificación de requerimientos generales.

- Herramientas a utilizar en el desarrollo.
- Objetivos del Proyecto.
- Estructura del Proyecto.
- Fechas estimadas de inicio y culminación de actividades.

Dichos autores referidos por Flores y Rufino (2007), también hacen énfasis a que este documento es la base para realizar los mencionados incrementos planificatorios, los cuales son más pequeños y acertados, ajustándose más a la realidad, por la imprecisión que puede tener una planificación para varios meses.

- **DOCUMENTOS DE ALCANCE Y REQUERIMIENTOS**

De acuerdo a Salazar y Prado-Calderón (2013) el objetivo del estudio indicará la finalidad a alcanzar, y para Mendoza y Reynoso (2001) citados por Flores y Rufino (2007), en toda metodología evolutiva incremental es importante tener claro el objetivo: los requerimientos, donde Cervantes y Gómez (2012) ratifican que al ser una parte esencial del desarrollo, debe ser atendida con el debido cuidado y esfuerzo.

Tales autores mencionados por Flores y Rufino (2007), también especifican que para la realización del documento de Alcance y Requerimientos del sistema, se debe recopilar la información siguiendo un esquema de trabajo básico, basado en la realización de entrevistas, reuniones y encuestas, con los diferentes grupos de usuarios involucrados en el proceso. Al final, este documento no solo promueve la calidad del proceso, sino también del producto, contemplando:

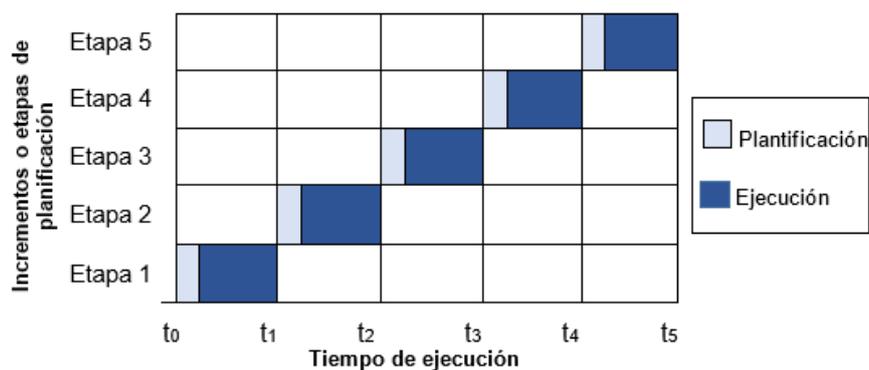
- Definición del problema.
- Especificación de los requerimientos.
- Alcance del desarrollo.

- **INCREMENTOS**

Una vez definido el objetivo utópico se tiene claro el fin pero no el camino, por lo cual interviene el aspecto más característico de la metodología: los incrementos, para establecer y lograr las metas a corto plazo. En este sentido, Mendoza y Reynoso (2001) mencionados por Flores y Rufino (2007) definen dos procesos dentro de los incrementos que son de suma importancia:

- **Planificación:** pretende estimar cuales actividades se realizarán dentro del próximo incremento, aquí es muy importante la administración de los recursos tanto cronológicos como psíquicos. Considerando lo expresado por Salazar y Prado-Calderón (2013) la planificación no cesa, e incluso al momento de la implementación, es necesario llevar a cabo acciones para adaptarse a las nuevas circunstancias.
- **Ejecución:** se llevan a cabo las actividades según lo planificado, se debe ser muy estricto en este sentido. La inversión de recursos empleados en una ejecución permite realizar una próxima planificación más exacta, esta es una razón por la que los primeros incrementos son más difíciles de planificar.

Benediktsson *et al.* (2006) mencionados por Mujumdar *et al.* (2012) hacen énfasis a los enfoques incrementales, destacando que son particularmente útiles cuando no se dispone de todo el personal necesario para completar el proyecto y cuando no es posible especificar completamente el producto requerido o formular plenamente el conjunto de expectativas. En la Figura 2. 12 se observa el esquema de la metodología bajo la Orientación al proceso.



**Figura 2. 12.** Esquema de la Orientación al proceso de MEIDAW.  
Fuente: Mendoza y Reynoso (2001) mencionados por Flores y Rufino (2007).

### • CONTROL DE INCREMENTOS

Esta metodología contiene dos aspectos muy importantes para el control en los incrementos:

- **Feedback:** Mujumdar *et al.* (2012) señalan que la retroalimentación (feedback) de una fase proporciona información de diseño para la siguiente fase, por lo que Mendoza y Reynoso (2001) mencionados por Flores y Rufino (2007) destacan que la metodología permite evitar las desviaciones desfavorables (feedback negativo), así como promover las favorables (feedback positivo).
- **Feedforwards:** Mendoza y Reynoso (2001) citados por Flores y Rufino (2007) indican que son de utilidad para evitar los obstáculos, permitiendo cambiar el rumbo (plan) en cualquier momento (incremento).

### • INFORME DE AVANCES

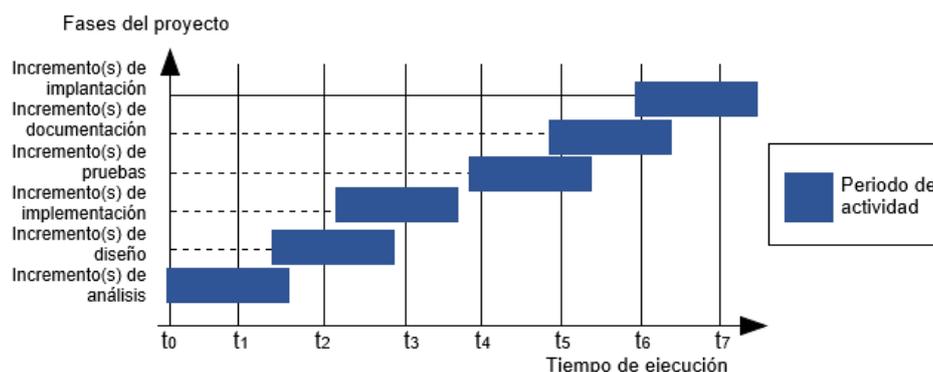
Mendoza y Reynoso (2001) referidos por Flores y Rufino (2007) también especifican que la planificación y las actividades de los incrementos planificatorios se deben registrar en los informes de avance, indicando:

- **Estado Actual del Sistema:** situación del sistema y de coherencia a las actividades descritas a continuación.

- **Actividades Completadas:** actividades terminadas en el período y los aspectos más importantes de su realización.
- **Actividades Retrasadas:** aquellas que no pudieron ser concretadas y las causas del retraso. Es de gran utilidad para mejorar las estimaciones.
- **Actividades Pautadas para los Próximos 15 días:** permite realizar la planificación como tal; este es el aspecto más importante del documento, ya que una planificación mal realizada puede causar descontrol.

#### 2.4.1.2. LA ORIENTACIÓN AL PRODUCTO

Para alcanzar la calidad de un sistema de Workflow existen varios aspectos que se deben considerar. Mendoza y Reynoso (2001) señalados por Flores y Rufino (2007) establecen un conjunto de actividades que permiten constituir definiciones y reglas para explotar el potencial del Workflow cuando se aplica a un caso particular. Existen aspectos que son de mayor importancia en este enfoque, por lo que dichos autores los definen como los distintos productos que se obtienen en cada una de esas macro-etapas (diferentes a los incrementos planificatorios), los cuales se describen a continuación (Figura 2. 13)



**Figura 2. 13.** Esquema de la Orientación al Producto de MEIDAW.  
Fuente: Mendoza y Reynoso (2001) citados por Flores y Rufino (2007).

#### • ANÁLISIS DEL SISTEMA

En esta etapa se determina el dominio de aplicación y los servicios que debe proporcionar el sistema, así como sus restricciones, de acuerdo a lo expuesto por Sommerville (2005) mencionado por Cervantes *et al.* (2012). Con base a esta información, Mendoza y Reynoso (2001) citados por Flores y Rufino

(2007) indican que al final de esta fase, se obtiene el Documento de Alcance y Requerimientos. Además, ellos coinciden en que debe ir enfocado a conocer en detalle las necesidades de los usuarios, por tal razón la intervención de los mismos es esencial en esta etapa.

- **DISEÑO DETALLADO DEL SISTEMA**

Al referirse a esta etapa, Mendoza y Reynoso (2001) citados por Flores y Rufino (2007) puntualizan que define el comportamiento y la estructura del sistema. Por su parte, CENDITEL (2013) enfatiza que se la considera una práctica importante tanto para la comprensión del problema para desarrollar el código fuente, como para la comprensión del código una vez desarrollado.

Al profundizar en estas consideraciones, dichos autores citados por Flores y Rufino (2007) señalan que en MEIDAW el diseño detallado consiste en la aplicación de especializados conceptos de diseño de sistemas de Workflow al resultado del análisis de requerimientos, para facilitar que el sistema satisfaga los requerimientos de los procesos de negocio involucrados y de las interfaces de usuario. Además, especifican los tópicos principales que debe contener el documento Diseño Detallado, que se constituye como el instrumento final de esta fase:

- **Reglas del Negocio:** condiciones que se deben satisfacer cuando se realiza una actividad de negocio, son políticas organizacionales que controlan los procesos de negocio, para el caso de estudio estas reglas se especifican en el diseño de cada proceso.
- **Flujo de Trabajo:** descripción de los procesos involucrados.
- **Roles:** personas y/o unidades responsables de las diferentes actividades que conforman los procesos de negocio; esta descripción se realiza por medio de la Tabla de Roles.
- **Estados y transiciones de un ítem de trabajo:** nombre y significado de cada estado y diagrama de transición de estados. Estos elementos se muestran en una tabla llamada Tabla de Transiciones.

- **Escalaciones:** condiciones que deben ser chequeadas periódicamente y/o que requieren la intervención de unidades superiores.
- **Notificaciones:** mensajes que envía el sistema a los usuarios, estos se basan generalmente en el incumplimiento de condiciones normales de flujo.
- **Interfaces:** pantallas del sistema y su interacción con los usuarios.
- **Reportes:** definición de documentos generados por el sistema para fines estadísticos y de control.

Para alcanzar la realización de tales tópicos, Flores y Rufino (2007) definen las siguientes actividades:

- Selección de la plataforma y tecnologías de desarrollo.
- Identificación de actividades.
  - Módulos que deben lanzar.
  - Acciones automatizadas a realizar.
  - Información a presentar al usuario.
  - Diseño de pantalla.
- Identificación de eventos.
  - Módulos que deben originar eventos.
  - Análisis de eventos automáticos.
- Aspectos de seguridad.
- Herramientas de control y umbral de escalado.

#### • **DESARROLLO DEL SISTEMA**

Mendoza y Reynoso (2001) señalados por Flores y Rufino (2007) especifican un conjunto de actividades que deben realizarse en esta etapa; la instalación y configuración de la arquitectura en la plataforma requerida, para que cada uno de los elementos identificados en las etapas de Análisis de Requerimientos y Diseño Detallado sean implementados, siguiendo la planificación fijada al comienzo de cada incremento. Al reflexionar en este sentido, Munassar y Govardhan (2010) mencionados por Mujumdar *et al.* (2012) también la

denominan fase de construcción, ya que consiste en la ejecución de lo establecido en la etapa de planificación.

Las actividades del desarrollo definidas por tales autores citados por Flores y Rufino (2007) se realizan en el siguiente orden:

1. **Configuración de la arquitectura de desarrollo:** instalación y configuración de herramientas y del servidor.
2. **Implementación de la arquitectura de datos:** Creación de las tablas, stored procedures, strings de conexión y demás elementos.
3. **Desarrollo de componentes y servicios:** Implementación de los elementos de la capa intermedia que interactúan sobre la base de datos y los repositorios de datos externos.
4. **Programación de pantallas, filtros y enlaces activos:** Construcción del front-end del sistema e implementación de la interfaz.
5. **Programación de notificaciones, escalaciones y reportes:** implementación de los elementos server-centric (acciones cuya condición de ejecución o disparador es dada por el motor de Workflow) y time-centric (acciones cuya condición de ejecución o disparador es dada por el paso del tiempo) que constituyen el sistema. También se implementan los reportes del sistema, según lo establecido en el diseño detallado del sistema.

- **PRUEBAS DEL SISTEMA**

Esta etapa constituye la verificación de que los procesos y actividades del sistema cumplen con las reglas del negocio establecidas según el diseño, de acuerdo a Mendoza y Reynoso (2001) citados por Flores y Rufino (2007). Por su parte, Trivedi (2012) enfatiza que las pruebas de software son más que una simple detección de errores; el software evaluado está operando bajo condiciones controladas, (1) para verificar que se comporta como se espera, (2) para detectar errores y (3) para validar que las correcciones del sistema son correctas. Las etapas se detallan a continuación:

1. **Verificación:** comprobación del software, para la conformidad y consistencia, evaluando los resultados en función de requisitos pre-especificados.
2. **Detección de errores:** en las pruebas se deben intentar hacer que las cosas salgan mal, para determinar si el sistema responde correctamente.
3. **Validación:** examina la corrección brindada por el sistema.

Para poder evaluar el correcto funcionamiento del sistema y llevar a cabo la verificación de los posibles errores encontrados, se consideran los siguientes tipos de pruebas:

- **Pruebas Unitarias:** Consisten en la verificación del funcionamiento de cada requerimiento funcionales, teniendo en cuenta cada una de las especificaciones necesarias que su correcta ejecución.
- **Pruebas de Regresión:** Consiste en la retroalimentación o verificación de los recursos que hayan presentado errores durante las pruebas unitarias
- **Pruebas Funcionales:** verifican que los elementos de interfaz y las actividades, transiciones y cálculos que realiza el sistema, cumplen con las funcionalidades preestablecidas.

Luego de que se ha evaluado el sistema, dependiendo de los resultados obtenidos, se puede volver a las etapas anteriores. Pero, en consonancia con Sommerville (2005) citado por Cervantes *et al.* (2012) cuando las pruebas del sistema completo son exitosas, está preparado para ser entregado al cliente.

- **DOCUMENTOS FORMALES DEL SISTEMA**

En esta etapa se elaboran los documentos que contienen las especificaciones de uso y administración del sistema, los cuales se definen a continuación.

- **Manual de usuario:** De acuerdo a Mendoza y Reynoso (2001) mencionados por Flores y Rufino (2007) presenta a los usuarios los

procedimientos necesarios para registrar, procesar y dar seguimiento a cada uno de los flujos que provee el sistema. Reflexionando en este sentido, CENDITEL (2013) y Galvão (2012) lo puntualizan como el documento donde se explica a los usuarios el uso del software.

- **Manual del administrador:** Según los autores mencionados por Flores y Rufino (2007), presenta toda la información técnica referente a la aplicación, donde se incluye la descripción general del sistema y se describe brevemente el propósito general del sistema. Y como lo indica Galvão (2012), este documento está destinado para que los desarrolladores puedan entender el sistema documentado. Por lo cual, se añade la Arquitectura de Datos, Arquitectura Funcional, Consideraciones de Seguridad, Instalación del sistema, etc.

Es importante recalcar que el Manual del administrador ha sido desarrollado bajo la etiqueta de Manual de programador, dado que es el nombre definido por los miembros de la entidad encargada de verificar el correcto funcionamiento del sistema y de brindar el posterior mantenimiento.

- **IMPLEMENTACIÓN DEL SISTEMA**

DSDM (2002) referido por Rivadeneira (2013) señala que en esta etapa el sistema es transferido al ambiente de producción y los usuarios son capacitados en el manejo del mismo, luego de que ha sido desarrollado y evaluado, de acuerdo a lo expuesto por Sommerville (2005) citado por Cervantes *et al.* (2012).

Para alcanzar esta última etapa de MEIDAW, Mendoza y Reynoso (2001) referidos por Flores y Rufino (2007), definen las actividades requeridas para la transición del sistema, detallando:

- **Configuración del Servidor de Producción y de la Base de datos:** instalación y configuración los servidores mencionados.

- **Migración del Sistema:** transferencia de la aplicación desde el servidor de desarrollo al servidor de producción, si estos servidores son diferentes.
- **Pruebas del Sistema en el Ambiente de Producción:** en dicho ambiente, se realizan pruebas funcionales y de carga de datos.

## **2.5. INTEROPERATIVIDAD, API E INTERFACES DE SISTEMAS**

Un aspecto relevante para el desarrollo de este proyecto es elegir los mejores métodos para permitan que el software sea interoperable, para lo cual Alayón (2016) enlista dos alternativas existentes: SOAP, que toman su nombre del protocolo de intercambios de mensajes que implementan: Simple Object Access Protocol (también llamadas web services) y REST, que es un acrónimo de Representational State Transfer. Luego de un análisis de las ventajas y desventajas de cada uno, se determina la utilización del último por las superioridades que representa frente a su contrario.

### **2.5.1. REST**

En primer lugar, Amodeo (2013) explica que REST no es una tecnología, ni siquiera una arquitectura, sino un estilo arquitectónico. Es un conjunto de restricciones a respetar cuando se diseña la arquitectura de los servicios web. Las restricciones propuestas por REST son las siguientes:

- Los servicios web no representan acciones, sino entidades de negocio.
- Cada recurso posee un identificador único universal (UUID o GUID) con el que se puede hacer referencia a él.
- La implementación, y la forma exacta en la que un recurso se representa internamente, debe ser privada y no accesible al exterior.
- Cada recurso tiene una interfaz, o conjunto de operaciones que admite. Basta saber el UUID del recurso para poder enviarle la operación que queremos realizar.
- La interfaz es homogénea para todos los recursos.

- Las operaciones son “stateless”. Es decir, el resultado de una operación es independiente de la conversación que hayan mantenido el cliente y el servidor anteriormente.
- Los recursos son multimedia, es decir, el estado de un recurso puede ser representado mediante distintos formatos. Por formato se entiende el formato concreto usado en la serialización del estado cuando se manda por red, por ejemplo: XML, JSON, imagen JPEG, imagen GIF, etc.

### **2.5.2. REST EN LA PRÁCTICA: HTTP**

Amodeo (2013) y Alayón (2016) destacan que es necesario utilizar un protocolo que cumpla los principios REST, y que esté ampliamente distribuido. Por lo cual, HTTP (Hyper Text Transfer Protocol) es una pieza fundamental en la que se basa la World Wide Web, y especifica como intercambiar entre cliente y servidor recursos web. Es un protocolo idóneo para implementar servicios web, ya que además de ser ubicuo, su diseño sigue los principios REST.

En la Figura 2. 14 se presenta un ejemplo de una petición HTTP donde se explica que en la primera línea se está usando una petición tipo POST contra la URI /server/payment. A continuación vienen las cabeceras con información adicional, como el tipo mime del cuerpo de la petición (Content-Type) o el tipo mime aceptable en la respuesta (Accept). Finalmente, tras una línea en blanco de separación, aparece el cuerpo, convenientemente formateado en el tipo mime especificado (application/x-www-form-urlencoded).

```

POST /server/payment HTTP/1.1
Host: www.myserver.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US, en; q=0.8
Cache-Control: max-age=0
Connection: keep-alive

orderId=34fry423&payment-method=visa&card-number=2345123423487648&sn=345

```

Figura 2. 14. Ejemplo de petición HTTP.

Fuente: Amodeo (2013)

Mientras que en la Figura 2. 15 se detalla una posible respuesta. En la primera línea sólo indica la versión del protocolo HTTP usado (HTTP/1.1) y el código de respuesta (201). Después aparecen las cabeceras, y tras una línea en blanco, el cuerpo de la respuesta.

```

HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
Location: https://www.myserver.com/services/payment/3432
Cache-Control: max-age=21600
Connection: close
Date: Mon, 23 Jul 2012 14:20:19 GMT
ETag: "2cc8-3e3073913b100"
Expires: Mon, 23 Jul 2012 20:20:19 GMT

{
  "id": "https://www.myserver.com/services/payment/3432",
  "status": "pending"
}

```

Figura 2. 15. Ejemplo de respuesta HTTP.

Fuente: Amodeo (2013)

### 2.5.3. URIS

Los Uniform Resource Identifiers o URIs son los identificadores globales de recursos en la web. Las URLs son uno de los dos tipos de URIs y se encargan de identificar un recurso de red mediante una IP o un DNS y tiene el siguiente formato (Cuadro 2. 1):

**Cuadro 2. 1.** Formato de una URL.

---

**Esquema de una URL** <esquema>:<parte específica esquema>/<ruta><querystring><fragmento>

---

Fuente: Amodeo (2013)

De acuerdo al formato, el esquema indica qué protocolo (http o https) se debe usar para acceder al recurso. Tras el esquema, viene la parte específica del esquema: la dirección del servidor donde se encuentra el recurso. Esta dirección se puede especificar mediante una dirección IP o mediante un nombre de dominio, y se separa del esquema mediante “//”. Opcionalmente se especifica el puerto a usar en la conexión concatenándolo con “:”. Esto no es necesario si se va a usar el puerto estándar (80 para http y 443 para https) del protocolo especificado. A continuación viene la ruta del recurso dentro del servidor: tiene la forma de un conjunto de segmentos separados por “/”. De forma opcional se tiene la query string, que indica parámetros adicionales de la URI a ser procesados en el servidor por la aplicación. Su uso más común es el de especificar criterios de búsqueda o filtrado. La query string se separa de la ruta mediante el carácter “?” y consiste en una serie de pares clave/valor separados por “&”. Dentro de cada par la clave y el valor se separan mediante “=”. Finalmente y también de forma opcional está el fragmento, que se separa del resto de la URI mediante “#”. Al contrario que la query string no se procesa en servidor, sino que está pensado para ser procesado por la aplicación cliente, no tiene impacto en el diseño de servicios web.

#### **2.5.4. VERBOS HTTP**

Una característica completamente alineada con REST del protocolo HTTP es el hecho de que tenga una interfaz uniforme para todos los recursos web. Alayón (2016) puntualiza que HTTP define un conjunto predefinido y cerrado de acciones o métodos HTTP. Por su parte, Amodeo (2013) matiza que es importante tener en cuenta que la propia especificación define los conceptos de seguridad e idempotencia, y clasifica los métodos conforme a estos dos criterios.

- Es seguro si no produce efectos secundarios. Por efecto secundario se entiende cualquier modificación del estado del servidor, o interacción de éste con cualquier otro sistema, que produzca efectos perceptibles por el

usuario. Normalmente sólo los métodos que representan lectura se consideran seguros.

- Es idempotente si la ejecución repetida de éste, con exactamente los mismos parámetros, tiene el mismo efecto que si sólo se hubiera ejecutado una vez. Esta propiedad permite reintentar con seguridad una petición una y otra vez, y tener la seguridad de que la operación no se va a duplicar.

En el Cuadro 2. 2 se presentan los verbos HTTP más comunes.

**Cuadro 2. 2.** Verbos HTTP más comunes.

Método	Seguro	Idempotente	Detalle
GET	Sí	Sí	Leer el estado del recurso.
PUT	No	Sí	Actualizar o crear.
DELETE	No	Sí	Eliminar un recurso.
POST	No	No	Cualquier acción genérica no idempotente.

Fuente: Amodeo (2013)

### 2.5.5. TIPO MIME

De acuerdo a Alayón (2016) otro aspecto muy apropiado para el desarrollo de servicios REST con el protocolo HTTP es su soporte para negociar distintos formatos (representaciones) a usar en la transferencia del estado entre servidor y cliente (y viceversa).

Por su parte, Amodeo (2013) especifica que en HTTP cada uno de estos formatos constituye lo que se llama un tipo MIME. Existe un directorio con tipos MIME estandarizados. Cada tipo MIME tiene el formato de <tipo>/<subtipo>. Algunos ejemplos de tipos MIME estándares: application/json, application/xml, application/atom+xml, application/javascript, text/html y audio/vorbis.

### 2.5.6. CÓDIGOS DE ESTADO

En HTTP el mensaje de respuesta contiene en su primera línea lo que se llama el código de estado, que indica el resultado de la operación. Los códigos de respuesta más usados son presentados por Amodeo (2013) en el cuadro:

**Cuadro 2. 3.** Códigos de respuestas más usados.

<b>Código</b>	<b>Explicación</b>
200	Indica éxito de la operación de forma genérica. Se usa cuando no hay otro código de éxito más específico. El mensaje de respuesta debe contener un cuerpo, en otro caso se usa 204.
201	Indica que se creó con éxito un nuevo recurso web. Suele devolverse cuando se ejecuta un método POST o PUT. En la respuesta se devuelve la URI del nuevo recurso creado dentro de la cabecera Location. La respuesta puede contener un cuerpo con los datos del nuevo recurso.
202	Indica que la petición se ha aceptado pero que no está completa. Se puede devolver en la respuesta un cuerpo con información sobre cuanto queda para que se complete la operación, o cómo monitorizar el progreso de ésta.
204	Indica éxito de la operación de forma genérica. Se usa cuando no hay otro código de éxito más específico. El mensaje de respuesta debe estar vacío y no tener cuerpo, en otro caso usar 200.
301	Indica que el recurso se ha movido a otra URI de forma permanente. La nueva URI se indica en la cabecera Location de la respuesta.
304	Indica que el recurso no se ha modificado. Normalmente es la respuesta de una cache a una operación. La respuesta no contiene cuerpo.
400	El mensaje de petición está mal formado
401	La petición no está autorizada, acceso denegado. Las credenciales de la petición son incorrectas o simplemente la petición no tenía credenciales.
403	Acceso denegado: las credenciales son insuficientes para acceder al recurso, el usuario no tiene permiso.
404	Recurso no encontrado.
405	Método no soportado, el verbo HTTP especificado en la petición no está implementada.
415	El tipo mime del cuerpo de la petición no está soportado por el servidor.
422	El cuerpo de la petición está mal formado. Por ejemplo un JSON o un XML con una sintaxis incorrecta (no cierran las llaves o las etiquetas), o tal vez el tipo mime especificado en la cabecera Content-Type no concuerda con el formato del cuerpo.
500	Se produjo un error inesperado.

**Fuente:** Amodeo (2013)

## **CAPÍTULO III. DESARROLLO METODOLÓGICO**

El desarrollo del núcleo de la plataforma de generación de aplicaciones de flujo de trabajo de los procesos administrativos, se llevó a cabo en la carrera de Computación de la ESPAM MFL, ubicada en el sitio El Limón de la ciudad de Calceta, cantón Bolívar – provincia de Manabí, el mismo tuvo una duración de 12 meses.

Para lo cual se utilizó MEIDAW, compuesta por dos fases. La primera estuvo orientada al proceso, mediante la realización del Plan de proyecto; documento de Alcance y Requerimientos; planificación, ejecución, control y documentación de los incrementos, considerando que cada incremento se ajusta más a la realidad, al ser más pequeño y acertado, permitiendo efectuar las correspondientes iteraciones de la siguiente fase de la metodología. La segunda fase fue orientada al producto, a través de las etapas: Análisis, Desarrollo, Pruebas, Documentos formales e Implementación del núcleo. Cabe recalcar que en esta fase no se implementaron ciertas actividades de la etapa Diseño detallado del sistema, porque no fue necesaria la realización de las mismas.

Además, se implementó el método bibliográfico para consultar a diferentes fuentes que sirvieron como base científica durante el desarrollo del proyecto, y el método analítico para analizar y comprender los diferentes elementos que intervienen en los procesos administrativos de los flujos de trabajo. Estos métodos son detallados durante la ejecución de las correspondientes actividades de la metodología, descritas a continuación.

### **3.1. MEIDAW**

La Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow propuesta por Mendoza y Reynoso (2001) mencionados por Flores y Rufino (2007), se utilizó para desarrollar el núcleo del Workflow que gestionará los procesos administrativos de la ESPAM MFL. Al ser una metodología

híbrida, fusiona elementos de la Metodología Evolutiva Incremental (MEI) para la orientación al proceso (fase 1), y el soporte de los estándares metodológicos de la WfMC para obtener la calidad del producto (fase 2).

### 3.1.1. FASE 1: ORIENTACIÓN AL PROCESO

#### 3.1.1.1. PLAN DE PROYECTO

Este documento provee una visión global del enfoque de desarrollo propuesto (Anexo 1) y se realizó a través de la ejecución de las siguientes actividades:

- **Identificación de requerimientos generales:** mediante el método analítico se analizaron los diversos elementos que pueden intervenir en los procesos administrativos (Figura 3. 1); comprendiendo el concepto y función de las tareas, actividades y participantes, además de las relaciones entre sí, y las respectivas instancias de los procesos.

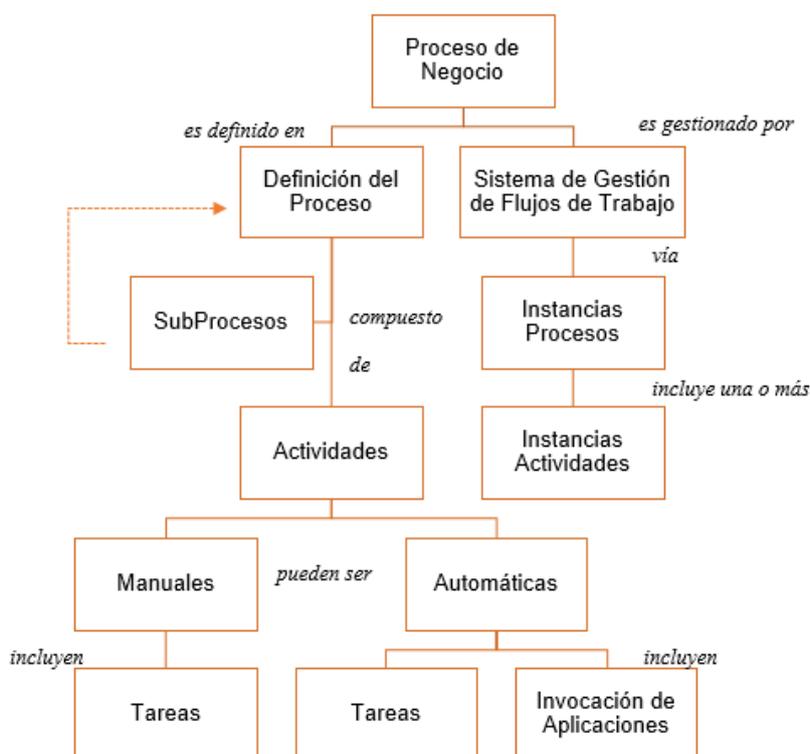


Figura 3. 1. Esquema general de la gestión de los procesos administrativos.

Fuente: Autores

- **Objetivo del proyecto:** el objetivo del núcleo es procesar la información de las actividades que intervienen en los diferentes procesos administrativos, para asignarlas a los recursos correspondientes, ya sean humanos o tecnológicos, a fin de que cumplan con las respectivas tareas: manuales o automáticas.
- **Definir el esquema de funcionamiento del núcleo:** a través del método bibliográfico, se investigó el modelo de referencia propuesto por la WfMC, los elementos que lo componen y la comunicación entre los mismos y se inquirió sobre el formato de los diferentes documentos que intervienen en dicha comunicación. Además, mediante una conversación informal con los integrantes de la UPS, se indagó las tecnologías utilizadas por dicho departamento, a fin de permitir que el sistema desarrollado siga los lineamientos del mismo.
- **Determinar las herramientas de desarrollo:** mediante el método bibliográfico, se obtuvo un índice de los lenguajes de programación más populares, el cual fue elaborado a partir de diversas variables, como por ejemplo, número de ingenieros cualificados en determinado lenguaje, las búsquedas que hacen los usuarios por información de los distintos lenguajes de programación, la demanda de cursos, o los lenguajes que están siendo más utilizados. Luego, mediante una tabla de comparación de las características más importantes, se analizaron los 3 lenguajes más destacables del índice mencionado, concluyéndose que Visual C# es el más adecuado para el desarrollo del núcleo. Las otras herramientas principales para el desarrollo, como IDE, SGBD y framework, fueron seleccionadas a partir del lenguaje escogido.
- **Estructura del proyecto:** detalla los diferentes documentos que fueron generados y utilizados durante el desarrollo de este trabajo, dividiéndolos en entregables. Estos fueron establecidos mediante el análisis de los documentos recomendados por MEIDAW, concluyéndose que todos, a excepción del Manual de Usuario, eran necesarios. La elaboración de dicho documento no fue relevante, porque no fueron usuarios comunes o finales quienes tuvieron que interpretar los recursos brindados por la WEB Api

desarrollada, sino las autoras de la tesis complementaria, encargada del diseño de las interfaces.

Además, mediante la reunión informal con la UPS, mencionada anteriormente, se determinó que también era necesario el Manual del programador, ya que permitirá que los integrantes de dicho departamento puedan brindar el posterior mantenimiento de la Web API, luego de la respectiva entrega por parte de los autores de este trabajo.

- **Definir las actividades, responsables y tiempos, estimados para el desarrollo:** De acuerdo a las habilidades de los integrantes, se realizó el cronograma de desarrollo.

### 3.1.1.2. DOCUMENTO DE ALCANCE Y REQUERIMIENTOS

Para la realización de este documento se efectuaron varias reuniones con el Ing. Joffre Moreira (tutor), el Ing. Javier López (colaborador directo) y las autoras de la tesis correspondiente al desarrollo de las interfaces del núcleo. Además, se contó con la capacitación de un experto en diseño y elaboración de herramientas Workflow, el Ing. Luis Ulloa, misma que fue dictada en el edificio de la Carrera de Computación de la ESPAM MFL. Se utilizó como modelo de referencia el estándar IEEE830 (Institute of Electrical and Electronics Engineers). El documento final, detallado en el Anexo 2, fue elaborado mediante las actividades descritas a continuación:

- **Definición del problema:** a fin de contribuir al Sistema de Información de la ESPAM MFL, es necesario que se desarrolle software para el área administrativa que pueda adaptarse a los cambios de los procesos o necesidades de los diferentes departamentos, y que además se integren en un solo sistema, para evitar inconvenientes al momento de realizar el intercambio de información.
- **Especificación de los requerimientos:** para definir los requerimientos funcionales y no funcionales del núcleo, se tomaron en cuenta los requerimientos generales definidos en el Plan de Proyecto (Anexo 1) y se

utilizó el método bibliográfico para investigar las funciones que debe cumplir el núcleo (motor) mediante el servicio de ejecución del Workflow, para gestionar todos los datos recibidos por las interfaces de usuario, de acuerdo a lo establecido en el modelo de referencia presentado por la WfMC.

- **Alcance del desarrollo:** elaborar el núcleo de un Workflow, según las especificaciones de los requerimientos generales (Anexo 1) y los requerimientos funcionales y no funcionales (Anexo 2), con una duración máxima de 12 meses.

### 3.1.1.3. INCREMENTOS PLANIFICADOS

Durante el desarrollo del núcleo, con el fin de definir las actividades a realizar y llevar un mayor control de las ejecutas, cada 15 días se efectuaron y documentaron los respectivos incrementos, característicos de la metodología aplicada. Estos incrementos están detallados en el Anexo 5 y están compuestos por las siguientes etapas:

- **Planificación:** se estimaron las tareas y actividades correspondientes a cada incremento, con base a lo detallado en el cronograma de actividades, indicando los recursos y tiempo asignados para la posterior ejecución. En cada incremento, esta etapa se hizo de forma paralela a la finalización del incremento anterior.
- **Ejecución:** se llevaron a cabo las actividades planificadas previamente, lo cual permitió alcanzar el desarrollo del núcleo.
- **Control de incrementos:** se efectuó la retroalimentación del desarrollo del núcleo, es decir, dependiendo de los resultados (positivos o negativos) de cada incremento, se tomaron las consideraciones necesarias que permitieron continuar con el propósito del proyecto.
- **Informes de avances:** se documentó la situación del núcleo, al finalizar cada incremento, indicando: estado actual, actividades completadas y retrasadas.

### 3.1.2. FASE 2: ORIENTACIÓN AL PRODUCTO

#### 3.1.2.1. ANÁLISIS DEL SISTEMA

El objetivo de esta fase fue definir los requerimientos funcionales y no funcionales necesarios para el desarrollo del núcleo del Workflow. Para lo cual, se realizaron las respectivas reuniones con el equipo de trabajo y los consultores externos. Esta etapa contempló la elaboración de los documentos presentados en la Figura 3. 2, que sirvieron como base para la elaboración del núcleo:



Figura 3. 2. Documentos obtenidos en el Análisis del sistema.

Fuente: Autores.

- **Plan de proyecto:** macro-planificación del proyecto (Anexo 1).
- **Documento de Alcance y Requerimientos:** constituye el documento más importante de esta etapa, basado en el estándar IEEE830. Especifica el listado de los requerimientos, dominio de la aplicación y servicios que debe proporcionar el núcleo (Anexo 2).
- **Casos de uso:** teniendo en cuenta los requerimientos funcionales estipulados en el documento de Alcance y Requerimientos, se procedió a realizar el diseño de los casos de uso mediante la herramienta de Office Visio 2013 (Anexo 3). Aunque este punto no ha sido establecido por la metodología empleada, a fin de brindar un documento que facilite la comprensión del núcleo desarrollado, desde el punto de vista del cliente final (quienes realizan las peticiones HTTP a la WEB API), se detalla el comportamiento del núcleo del Workflow. Se especifica mediante actores la participación e interacción de los diferentes roles de usuarios y el funcionamiento de los procesos del sistema.

### 3.1.2.2. DISEÑO DETALLADO DEL SISTEMA

Tal como se menciona al inicio de este capítulo, durante el desarrollo del núcleo no se consideró necesaria la implementación de ciertas actividades de esta etapa. A continuación, se detalla cada punto que formó parte de esta fase, de acuerdo a lo especificado por MEIDAW:

- **Reglas del Negocio:** el núcleo permite que la ESPAM MFL pueda gestionar las reglas que controlan sus procesos administrativos en el momento que lo crea conveniente, por lo cual no fue necesario definir las durante el desarrollo. Para el óptimo funcionamiento del núcleo, dichas reglas deberán ser implementadas con base a lo estipulado en el Manual de Procesos de la institución.
- **Flujo de Trabajo:** dado que no se automatizó un proceso en concreto, los flujos de los procesos administrativos de la ESPAM MFL podrán ser automatizados en cualquier momento, mediante el núcleo desarrollado.
- **Roles:** se determinaron ciertos roles (Cuadro 4. 9) que podrán ser utilizadas para alcanzar el modelado y supervisión de los procesos según lo definido en el esquema general de funcionamiento (Figura 4. 1). Además, los autores analizaron la existencia de muchos cargos administrativos que no pudieron ser definidos durante el desarrollo del núcleo, por lo cual se estableció un rol que facilitara que la definición de tales cargos y sus responsables, puedan ser establecidos en cualquier momento, permitiendo así que el núcleo sea flexible a los cambios presentados en el departamento administrativo.
- **Estados y transiciones de un ítem de trabajo:** dentro de un flujo de trabajo, el núcleo definirá el estado de las instancias de los procesos administrativos y sus actividades, de las notificaciones enviadas por el sistema a los usuarios (Administrador, Supervisor, DBA, Modelador y diseñador) responsables de las tareas durante el diseño de los procesos y a los usuarios Operadores encargados de los trámites durante la ejecución de los mismos. En el Cuadro 3. 1 se detalla la nomenclatura de los

posibles estados de los elementos mencionados y su respectiva descripción

**Cuadro 3. 1.** Posibles estados de los elementos del flujo de proceso.

Elemento	Nomenclatura	Alerta	Ruta	Estado	Descripción
<b>Instancia</b>	J			En ejecución	Indica que la instancia del proceso sigue el flujo normal.
	E			En espera	Indica que dentro del flujo hay una actividad de llamada a otro proceso, ejecutándose.
	F			Finalizado	Indica que la instancia del proceso ha terminado exitosamente.
	T			Terminado	Indica que la instancia del proceso ha terminado de forma inesperada.
	D			Eliminado	Elimina una instancia.
<b>Archivo</b>	N			Nuevo	Indica que el archivo es almacenado por primera vez.
	A			Actualizado	Indica que es una nueva versión de un archivo existente.
	U			Último	Indica que es la última versión del archivo
<b>Tarea del proceso (Trámite)</b>	I			Inicial	Indica el estado inicial de una tarea, cuando se instancia la versión de un proceso.
	E			Ejecución	Indica que se ha notificado al Operador el inicio de la ejecución de la tarea.
	F			Finalizada	Indica que la tarea ha terminado exitosamente.
	T			Terminada	Indica que la tarea no pudo ser completada por errores (fatales) durante la ejecución.
	A			Atrasada	Indica que la tarea no fue ejecutada por el usuario en el tiempo estimado.
<b>Tarea de la estación</b>	E	-	-	Enviada	Indica que la tarea ha sido enviada al usuario.
	L	-	-	Leída	Indica que la tarea ha sido leída, pero no iniciada por el usuario.
	T	-	-	Finalizada	Indica que la tarea ha sido finalizada exitosamente por el usuario.
	AC	-	-	Aceptada	Indica que el administrador ha revisado y aceptado la tarea
	RE	-	-	Rechazada	Indica que el administrador ha revisado y rechazado la tarea
	A	-	-	Atrasada	Indica que la tarea no fue ejecutada por el usuario en el tiempo estimado por el Administrador del proceso.
<b>Mensaje de La Tarea</b>	E	No	Si	Enviado	Indica que el sistema ha enviado un mensaje a un usuario.
	R	No	Si	Recibido	Indica que el mensaje ha sido recibido por el usuario.
	L	No	Si	Leído	Indica que el usuario ha leído el mensaje.

	L (Revisión)	No	No	Leído	Indica que el usuario ha enviado la tarea para su revisión.
	EL	Si	No	Eliminado	Indica que el mensaje y/o la tarea han sido eliminados
	RA	Si	No	Reasignado	Indica que el usuario responsable de la tarea correspondiente al mensaje ha sido cambiado
	T	No	No	Terminado	Indica que el Administrador ha aceptado la tarea.
<b>Mensaje Del Trámite</b>	E	No	Si	Enviado	Indica que el sistema ha enviado un mensaje a un usuario.
	R	No	Si	Recibido	Indica que el mensaje ha sido recibido por el usuario.
	L	No	Si	Leído	Indica que el usuario ha leído el mensaje.
	A	SI	SI	Atrasado	Indica que la tarea está atrasada.
	EL	Si	No	Eliminado	Indica que el mensaje ha sido eliminado.
	T	No	No	Terminado	Indica que el Usuario ha terminado la tarea.
	D	No	Si	Documento	Indica que el mensaje es una tarea automática
	ER	SI	NO	Error	Indica que ha ocurrido un error durante la ejecución de la tarea.

Fuente: Autores.

En la Figura 3. 3 se detallan los estados que puede tener un proceso administrativo durante su ejecución o instancia, mismo que contiene trámites (actividades) realizadas por los operadores o usuarios del proceso, por lo tanto, en la Figura 3. 4 se presentan las posibles fases de dichos trámites. Cabe recalcar, que se establece como sistema de comunicación el envío y recepción de notificaciones y alertas entre los participantes durante la ejecución de procesos (Figura 3. 5).

Antes de que un proceso pueda ser ejecutado, se debe llevar a cabo una serie de tareas para su diseño o preparación, incluyendo la asignación de recursos (tiempo, usuario responsable, formulario, etc.), para lo cual se presenta en la Figura 3. 6 los estados y transición de las tareas realizadas por los usuarios para la creación de los procesos administrativos. Finalmente, en la Figura 3. 7 se detalla la forma de comunicación entre el sistema y los responsables de dichas tareas.

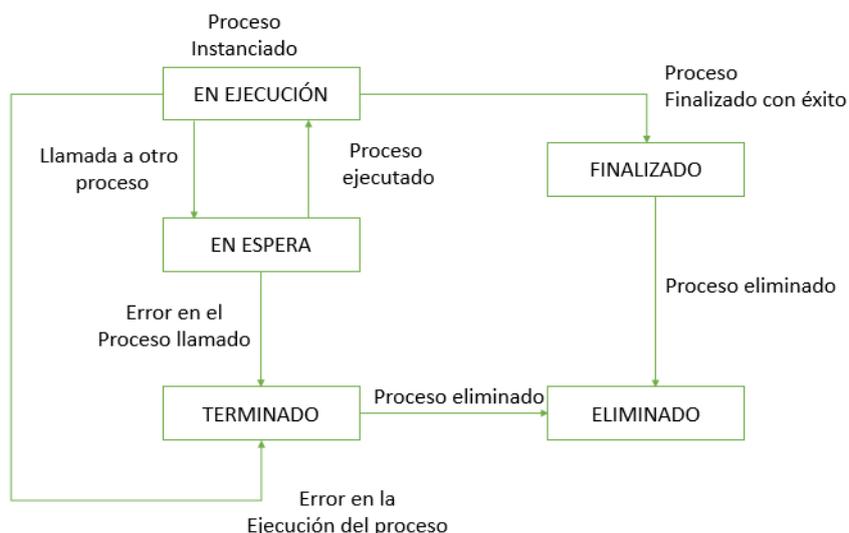


Figura 3. 3. Estados y transición durante la ejecución de un proceso administrativo.

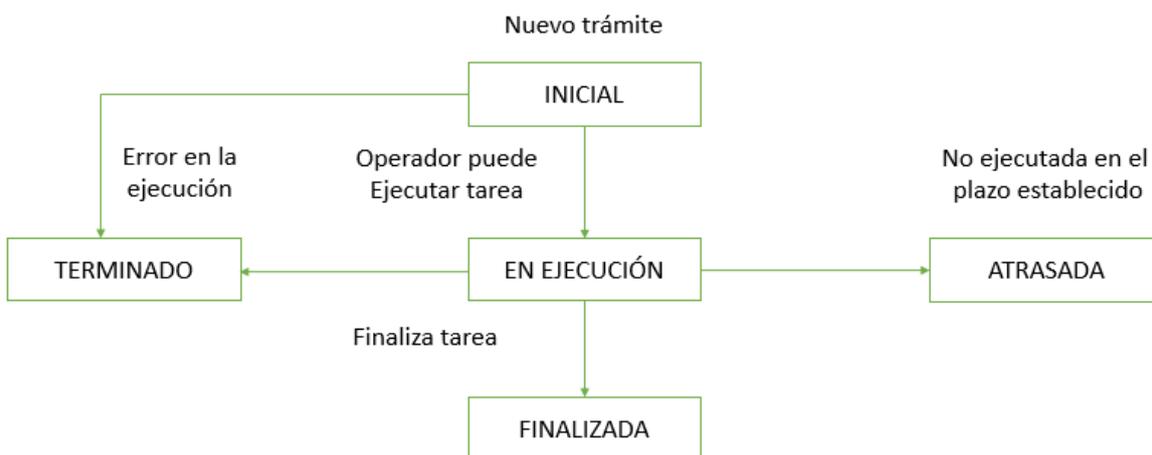


Figura 3. 4. Estados y transición de las tareas o trámites durante la ejecución de un proceso.

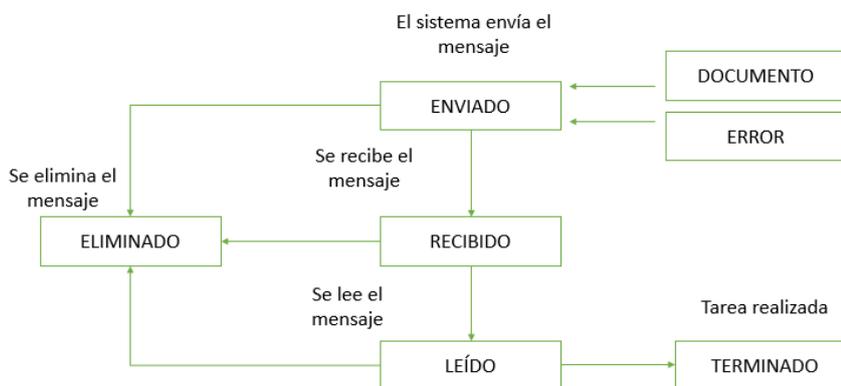
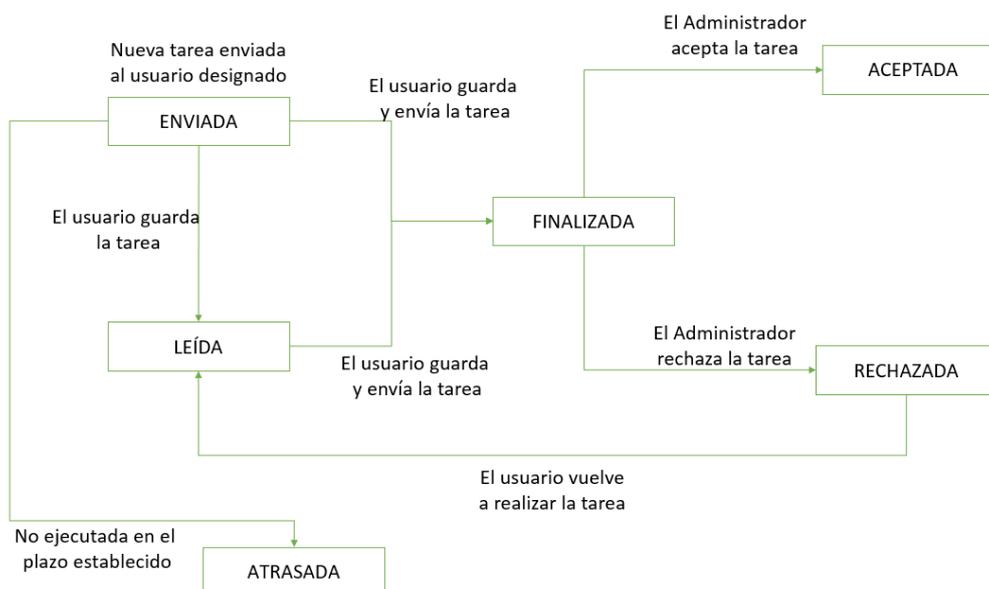
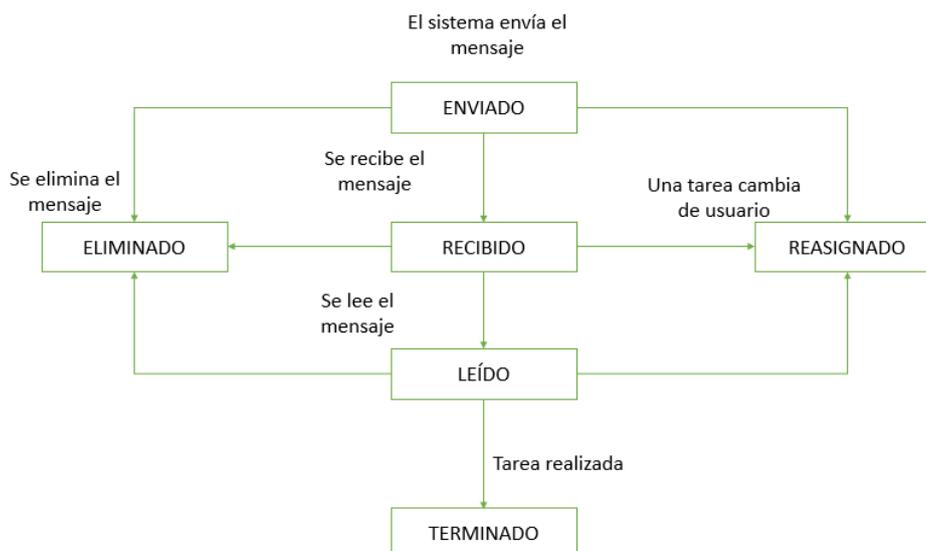


Figura 3. 5. Estados y transición de la comunicación (notificaciones y alertas) entre los usuarios, durante la ejecución del proceso.



**Figura 3. 6.** Estados y transición de la ejecución de las tareas durante la creación o diseño del proceso.



**Figura 3. 7.** Estados y transición de los mensajes enviados a los usuarios, durante el diseño del proceso.

- **Escalaciones:** el núcleo se encargará de chequear periódicamente las condiciones de tiempo establecidas en las tareas durante el diseño y ejecución de los procesos. Esto se lo realizará mediante la utilización de la herramienta SQL denominada SQL Server Agent.
- **Notificaciones:** el Centro de Mensajes se encuentra dividido de la siguiente manera: Una tabla para los mensajes de las tareas de definición del proceso y otra tabla para los mensajes que se crean durante la ejecución

de un proceso, en la Figura 4. 7 se detalla el diagrama entidad-relación encargado de gestionar dichos mensajes y notificaciones. De esta forma, el núcleo se encarga de almacenar las notificaciones enviadas a los usuarios correspondientes en cuanto a los diferentes estados de las tareas: inicio, eliminación, reasignación, atraso y cumplimiento.

- **Interfaces:** 4 de las 5 interfaces presentadas en el Modelo de Referencia propuesto por la WfMC, fueron realizadas por las autoras de la tesis complementaria, para su respectiva integración con el núcleo de este proyecto.
- **Reportes:** el sistema podrá generar las estadísticas de los tiempos de ejecución de cada proceso en sus diferentes instancias, detallando cada una de sus actividades. También podrá realizar reportes por responsables indicando las tareas a su cargo, teniendo en cuenta el tiempo que tardó en realizarlas. Esto se realiza con el fin de que la ESPAM MFL lleve un control de sus procesos administrativos, facilitando la toma de decisiones.

Las actividades que se llevaron a cabo durante esta etapa son detalladas a continuación:

- a) Selección de la plataforma y tecnologías de desarrollo:** Esta actividad fue realizada durante la elaboración del Plan de Proyecto (Anexo 1- Capítulo 5, sección Pila para el desarrollo), donde se especifican las tecnologías utilizadas. A continuación, en el Cuadro 3. 2 se presenta un resumen de las mismas.

**Cuadro 3. 2.** Resumen y descripción de las plataformas y tecnologías de desarrollo.

#	Tipo de tecnología	Tecnología implementada	Descripción
1	SGBD	SQL Server 2008.	Para el diseño e implementación de la arquitectura de datos y procedimientos almacenados
2	IDE	Visual Studio 2015	Para el desarrollo de la capa Negocio, mediante la Web API destinada para que la creación de servicios HTTP que permitieron el desempeño óptimo del núcleo del Workflow.
	Framework de desarrollo	WEB API REST ASP.NET	
	Lenguaje de programación	Visual C#	
	Servidor web	Internet Information Service (IIS)	
3	Cron Job	Sql Server Agent	Para facilitar la verificación de forma periódica del cumplimiento de las actividades que intervendrán en los diferentes flujos de trabajos instanciados por el núcleo.
4	Framework de documentación	Swagger UI	Para la generación automática de documentación y para la ejecución de las pruebas que permitieron verificar el funcionamiento correcto del núcleo del Workflow.

Fuente: Autores

**b) Identificación de actividades:** Cada actividad con sus responsables y tiempo de duración, que se llevaron a cabo para alcanzar el desarrollo del núcleo del Workflow están especificados en el cronograma de desarrollo presentado en el Cuadro 4. 6 y en el Anexo 1 de este documento. Por su parte, el Cuadro 3. 3 corresponde a las actividades que componen la parte esencial del núcleo, clasificadas de acuerdo a lo estipulado en la metodología.

**Cuadro 3. 3.** Identificación de actividades.

Módulos realizados	Tomando en cuenta el modelo de referencia en la elaboración de Workflow propuesto por la WfMC, se ha desarrollado principalmente: <ul style="list-style-type: none"> <li>• <b>Motor o núcleo del Workflow:</b> proporciona el control del ambiente de ejecución de las instancias de flujos de trabajo de los procesos administrativos de la ESPAM MFL.</li> <li>• <b>Servicio de ejecución de Workflow:</b> interpreta la descripción de procesos administrativos y controla las diferentes instancias de los mismos, secuencia de actividades, adiciona elementos a la lista de trabajo de los usuarios, e invoca aplicaciones necesarias.</li> </ul>				
Acciones automatizadas realizadas	De acuerdo a los requerimientos funcionales (RF) establecidos en el documento de Alcance y Requerimientos (presentado en resumen en la sección de Resultados y detallado por completo en el Anexo 2), se determinaron las acciones principales que permitieron que el núcleo satisfaga las necesidades de las interfaces comprendidas en el modelo de referencia de la WfMC, desarrolladas por las autoras de la tesis complementaria: <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Interfaz de Programación de Aplicaciones de Workflow (WAPI)</th> <th style="text-align: center;">Acciones automatizadas</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> <li>• Herramientas de definición de procesos.</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>• Definición de tareas para la creación de un proceso.</li> <li>• Supervisión de tareas para diseño del</li> </ul> </td> </tr> </tbody> </table>	Interfaz de Programación de Aplicaciones de Workflow (WAPI)	Acciones automatizadas	<ul style="list-style-type: none"> <li>• Herramientas de definición de procesos.</li> </ul>	<ul style="list-style-type: none"> <li>• Definición de tareas para la creación de un proceso.</li> <li>• Supervisión de tareas para diseño del</li> </ul>
Interfaz de Programación de Aplicaciones de Workflow (WAPI)	Acciones automatizadas				
<ul style="list-style-type: none"> <li>• Herramientas de definición de procesos.</li> </ul>	<ul style="list-style-type: none"> <li>• Definición de tareas para la creación de un proceso.</li> <li>• Supervisión de tareas para diseño del</li> </ul>				

	<ul style="list-style-type: none"> <li>proceso.</li> <li>Definición del Flujo (diagrama).</li> <li>Diseñador de Formularios.</li> <li>Integración Interna (II)</li> <li>Integración Externa (IE)</li> <li>Vista de datos</li> <li>Reporte de formularios.</li> <li>Aprobar o rechazar tareas durante el diseño del proceso.</li> <li>Definición de un proceso.</li> <li>Asignación de recursos al diagrama.</li> <li>Rediseño de un proceso.</li> </ul>
<ul style="list-style-type: none"> <li>Aplicaciones clientes</li> </ul>	<ul style="list-style-type: none"> <li>Instancia de un proceso.</li> <li>Historial de instancia de un proceso.</li> <li>Ejecución de las actividades.</li> <li>Gestión de cargos administrativos.</li> <li>Definición de participantes.</li> </ul>
<ul style="list-style-type: none"> <li>Aplicaciones Invocadas</li> </ul>	<ul style="list-style-type: none"> <li>Integración de datos.</li> </ul>
<ul style="list-style-type: none"> <li>Herramientas de administración y monitoreo</li> </ul>	<ul style="list-style-type: none"> <li>Reiniciar actividad.</li> <li>Reasignación de participantes.</li> <li>Errores durante la ejecución de un proceso.</li> <li>Supervisión de tareas (trámites) del proceso.</li> <li>Mecanismos de control.</li> </ul>
Información a presentar al usuario	<p>La información más importante que será presentada a los usuarios de las instancias de los procesos administrativos, será realizada en las siguientes acciones, también indicados como requerimientos funcionales del documento de Alcance y Requerimientos, mencionado anteriormente:</p> <ul style="list-style-type: none"> <li>Notificaciones.</li> <li>Reportes.</li> </ul>
Diseño de interfaces	Esta actividad fue realizada por las autoras de la tesis complementaria, encargadas de las interfaces del Workflow.

Fuente: Autores

**c) Identificación de eventos:** En el Cuadro 3. 4 se definieron los requerimientos funcionales (RF) que originan eventos, ya sean manuales o automáticos y se detalla el análisis de la naturaleza de los mismos.

**Cuadro 3. 4.** Identificación de eventos.

<b>Cód.</b>	<b>Requerimiento Funcional</b>	<b>Origina eventos</b>	<b>Tipo de evento</b>	<b>Análisis o explicación</b>
<b>RF01</b>	Definición de tareas para la creación de un proceso.	✓	Manual y Automático	Es necesaria la intervención del Administrador para el envío de tareas (formularios, validaciones, diagrama, etc.) a los participantes responsables. Posteriormente, el core se encarga de capturar los datos, guardando las tareas en la BD y enviando un mensaje a cada usuario asignado, con la tarea respectiva.
<b>RF02</b>	Supervisión de tareas para diseño del proceso.	✓	Manual y Automático	Es necesaria la intervención del Supervisor para inspeccionar el estado de las tareas de los diversos procesos que estén a su cargo. El core automáticamente notifica al Supervisor sobre las tareas no completadas.
<b>RF03</b>	Definición del Flujo (diagrama)	✓	Manual y Automático	Es necesaria la intervención del Modelador para la creación del diagrama de un proceso. El core automáticamente notifica al administrador el cumplimiento de la actividad.
<b>RF04</b>	Diseñador de Formularios.	✓	Manual y Automático	El Diseñador crea los formularios que se requieran para la ejecución de un proceso. El core automáticamente notifica al administrador el cumplimiento de la actividad.
<b>RF05</b>	Integración Interna (II)	✓	Manual y Automático	El DBA especifica los campos de las consultas a los formularios ingresados por los Operadores. El core automáticamente notifica al administrador el cumplimiento de la actividad.
<b>RF06</b>	Integración Externa (IE)	✓	Manual y Automático	El DBA puede consultar información a sistemas externos mediante servicios web, indicando la ruta y los parámetros de entrada y salida necesarios. El core automáticamente notifica al administrador el cumplimiento de la actividad.
<b>RF07</b>	Vista de datos	✓	Manual y Automático	El Diseñador puede crear consultas con diseño personalizado sobre la información registrada en el sistema. El core automáticamente notifica al administrador el cumplimiento de la actividad.
<b>RF08</b>	Reporte de formularios	✓	Manual y Automático	El Diseñador puede crear reportes personalizados en formato PDF. El core automáticamente notifica al administrador el cumplimiento de la actividad.
<b>RF09</b>	Aprobar o rechazar tareas durante el diseño del proceso.	✓	Manual y Automático	El Administrador puede revisar y ejecutar las tareas (Diagramas, Integraciones, etc.) realizadas por los demás usuarios y según ello, podrá aceptar o rechazar la tarea. El core automáticamente notifica al responsable sobre la decisión tomada por el Administrador. En caso de que se rechace la tarea, el Administrador deberá indicar el motivo y el Usuario tendrá que volver a hacer la tarea.
<b>RF10</b>	Definición de un proceso.	✓	Automático	El core se encargará de analizar y capturar desde el diagrama XML que haya guardado previamente el Modelador los elementos que intervengan en el proceso y los almacenará en las respectivas tablas de la base de datos.
<b>RF11</b>	Asignación de recursos al diagrama.	✓	Manual y Automático	Es necesaria la intervención del Administrador para la asignación de los respectivos recursos de una actividad, automáticamente el core procesará y enviará la información a la BD.

<b>RF12</b>	Instancia de un proceso.	✓	Manual y Automático	Para los flujos de procesos con eventos de inicio de mensaje, luego de que se cumpla la condición de inicio, se instanciará el proceso. Para los subprocesos, se iniciarán automáticamente después de la llamada del proceso respectivo. Para los de inicio normal, deberán ser instanciados por el Operador responsable del proceso.
<b>RF13</b>	Eliminar instancia de un proceso.	✓	Manual y Automático	En caso de que se presenten "errores fatales" durante la ejecución de los procesos, el Supervisor podrá dar de baja a la instancia. Automáticamente el core se encargará de eliminar todos los registros necesarios.
<b>RF14</b>	Reiniciar instancia de un proceso.	✓	Manual y Automático	En caso de que sea necesario, el Supervisor podrá reiniciar la instancia. Automáticamente el core eliminará todo los registros realizados hasta ese punto de la instancia.
<b>RF15</b>	Historial de instancia de un proceso.	✓	Automático	Cada vez que el estado de una instancia sufra cambios, se almacenará el respectivo registro en la BD, como mecanismo de control.
<b>RF16</b>	Rediseño de un proceso.	✓	Manual y Automático	Es necesaria la intervención del Administrador para el versionamiento. El core automáticamente analiza los elementos de la nueva versión y guarda en la BD.
<b>RF17</b>	Supervisión de tareas (trámite) del proceso.	✓	Manual y Automático	Es necesaria la intervención del Supervisor para que verifique la ejecución de las actividades o tareas del proceso administrativo. Si una tarea no es completada a tiempo, se notificará al Supervisor sobre lo sucedido.
<b>RF18</b>	Ejecución de las actividades.	✓	Manual y Automático	Las actividades manuales y de usuarios requieren del responsable mientras que las automáticas son iniciadas sin la intervención de los usuarios.
<b>RF19</b>	Reiniciar actividad.	✓	Manual y Automático	En caso de que un Operador se retrase en la realización de una actividad, el Supervisor podrá reiniciarla. Automáticamente el trámite volverá a su modo inicial.
<b>RF20</b>	Gestión de cargos administrativos.	✓	Manual	Es necesaria la intervención del Administrador para que gestione los cargos.
<b>RF21</b>	Definición de participantes.	✓	Manual	Es necesaria la intervención del Administrador para que designe los participantes que deberán cumplir con las actividades o tareas del proceso.
<b>RF22</b>	Reasignación de participantes.	✓	Manual	Es necesaria la intervención del Administrador para que reasigne los participantes que deberán cumplir con las actividades o tareas del proceso.
<b>RF23</b>	Mecanismos de control	✓	Automático	Mediante la ejecución del script correspondiente al cron job, se verifica de forma automática el cumplimiento de las actividades, en los tiempos especificados.
<b>RF24</b>	Errores durante la ejecución de un proceso.	✓	Automático	El core automáticamente detectará los errores presentados durante la ejecución del proceso y presentará al Supervisor las posibles alternativas de solución.
<b>RF25</b>	Notificaciones	✓	Automático	Se notifica automáticamente a cualquier Usuarios, en el momento oportuno, de acuerdo a las necesidades correspondientes.
<b>RF26</b>	Reportes	✓	Manual y	Es necesaria la intervención del Administrador para

			Automático	generar los reportes correspondientes. El core procesará la información necesaria para enviarla a las aplicaciones clientes.
RF27	Integración de datos	✓	Manual y Automática.	Es necesaria la intervención del DBA para que realice la escritura y ejecución de las integraciones. El core procesará la información necesaria para enviarla a las aplicaciones clientes.

Fuente: Autores

**d) Aspectos de seguridad:** como mecanismos de seguridad se implementaron los filtro de autenticación y autorización proporcionados por ASP.NET Web API2, que son descritos a continuación:

- **Filtro de autenticación:** comprende la autenticación de los usuarios registrados en la base de datos a través de OAuth 2.0, si las credenciales son correctas se le asigna un token con el que puede acceder a los recursos.
- **Filtro de autorización:** el núcleo estará compuesto de controladores los mismos que tendrán diversos recursos, los mismo que serán asignados a los usuarios correspondientes, para ello se verificará el rol respectivo, de acuerdo a lo establecido en la Base de Datos del sistema. Existen 3 tipos de autorizaciones:
  - **Authorize:** define el acceso a todos los usuarios autenticados en el sistema.
  - **Authorize Roles:** esta autorización permite acceder solo a los usuarios con roles autorizados, definidos como los tipos de usuario del sistema.
  - **Authorize Users:** Permite el acceso a los usuarios autorizados, sin definir el rol que tengan.

En este aspecto de seguridad interviene lo establecido en la Figura 4. 6, correspondiente al login y definición de tipos y características de los usuarios administrativos de la ESPAM MFL.

**e) Herramientas de control:**

Para el control del cumplimiento de las actividades que intervienen en un proceso se utiliza el agente de SQL con los denominados Jobs. Además, para

obtener acceso a los recursos (api), se cuenta con la autorización por roles, mediante el uso de tokens autogenerados.

Por otra parte, en la tabla de los usuarios del sistema (Cuadro 4. 9) se ha especificado el rol de Supervisor, permitiendo que este pueda revisar las actividades ejecutadas, faltantes y vencidas de cada proceso. A su vez, podrá realizar consultas y reportes sobre el progreso y estado actual del proceso administrativo que tiene a su cargo. Además, en caso de que un Operador no haya completado una actividad previamente asignada, se notificará directamente al Supervisor y al Administrador del proceso.

### **3.1.2.3. DESARROLLO DEL SISTEMA**

Durante esta etapa se llevaron a cabo todas las actividades establecidas en la planificación. Por lo cual, cada 15 días se realizaron reuniones entre los integrantes y el tutor del proyecto, a fin de llevar un control y retroalimentación de la ejecución de los incrementos planificados en la fase 1 de la metodología. A continuación, se detallan las principales actividades de desarrollo.

#### **1. Configuración de la arquitectura de desarrollo:**

Las herramientas establecidas mediante la elaboración del Plan de proyecto (Anexo 1- Capítulo 5, sección Pila para el desarrollo), y presentadas de forma resumida en el Cuadro 3. 2, fueron obtenidas de la siguiente forma: el IDE Visual Studio 2015 fue facilitado por la UPS de la institución, mismo que permitió de forma automática la configuración del servidor IIS para la codificación, implementación, hospedaje y administración del núcleo durante la etapa de desarrollo. Por su parte, el SGBD SQL Server 2008 R2 fue descargado desde el sitio web oficial de Microsoft.

Posteriormente, se procedió con la respectiva instalación. En primer lugar, se instaló el SGBD y el IDE en los equipos de los integrantes del proyecto. Luego, en VS se instaló el componente Swagger, que brindó una interfaz amigable

para la generación automática de la documentación de la API web y su vez facilitó la ejecución de las pruebas funcionales y de carga de datos.

Como última herramienta o complemento tenemos el Agente de SQL, el cual permite la verificación periódica de las fechas límite tanto de las tareas de diseño como de los trámites de las instancias de los procesos.

## **2. Implementación de la arquitectura de datos:**

Se realizó el diseño de la base de datos en SQL-Server 2008 R2, de acuerdo a los requerimientos funcionales (definidos en el Anexo 2, sección 3) presentados de forma resumida en el Cuadro 4. 13. Primero, se creó una nueva BD nombrada “Motor\_Workflow” y luego las entidades (tablas) correspondientes a los diferentes aspectos o enfoques de la base de datos. En este sentido, se establecieron los esquemas: diseño de proceso, encargado de contener las tablas para la creación y actualización (versionamiento) de los procesos, y las actividades de cada proceso; instancia de proceso, destinado a la ejecución de los procesos, luego de que han sido almacenados en la BD; tarea, establecido para gestionar las responsabilidades de los usuarios encargadas por el Administrador del proceso, cada vez que se requiera automatizar un nuevo proceso, por ejemplo: enviar al Diseñador de formularios los campos necesarios para la interfaz que debe crear; usuario, encargado de la gestión de los usuarios para la ejecución de los procesos y también para los responsables de crear nuevos procesos.

Una vez definidas las tablas, se crearon procedimientos almacenados (SP). Donde se estableció la lógica de las diferentes operaciones y transacciones realizadas por el núcleo. Esto se hizo con la finalidad de permitir agilizar los tiempos de respuestas de las peticiones de los clientes, constituyendo la comunicación desde el código de la API a la BD de forma sencilla y óptima.

Es importante mencionar que para la escritura de las entidades y los SP, se utilizó el estilo UpperCamelCase, mientras que para los campos de dichas entidades se hizo uso del estilo lowerCamelCase.

### **3. Desarrollo de componentes y servicios:**

Luego de definir toda la arquitectura de datos, se creó una solución en VS nombrada “motorWorkflow”, misma que fue publicada en el Team Foundation de Microsoft para facilitar la integración de código por parte de los integrantes, de forma remota. Luego, se creó la respectiva conexión con la BD mediante un elemento Identity y la inserción de los SP de la BD al IDE VS. Posteriormente, en la carpeta “Models” se crearon dos carpetas: “Entidades”, para añadir los campos y constructores de las entidades de la BD, y “Logic”, como capa intermedia para comunicación entre los SP de la BD y la API web en VS. Otra carpeta importante de mencionar es “Controllers”, la cual contiene los recursos necesarios para la comunicación entre el núcleo y las interfaces, dicha comunicación se la realiza mediante las peticiones HTTP: GET, POST, PUT y DELETE.

Una vez determinada la estructura de la solución, y establecida la conexión a la BD, se crearon las clases en la carpeta “Entidades”, mismas que representan las tablas de la BD, donde se registraron sus respectivas variables o propiedades. Cabe mencionar que para la escritura de las clases se siguió el estilo UpperCamelCase, mientras que las variables se escribieron con minúsculas y anteponiendo un guion bajo en cada palabra, ejemplo: `_id_proceso`.

A continuación, se procedió a agregar las clases de la carpeta “Logic” para crear los métodos encargados de conectar la solución en VS con los SP de SQL. Es decir, los métodos en C# almacenados en su respectiva clase Logic tuvieron un objeto encargado de realizar la conexión con los SP de la BD. Dado que son muchos métodos, no se incluirán en la documentación, pero para mayor comprensión del lector, se ubica como ejemplo el código SQL del SP

para crear las tareas de definición de un proceso (Figura 3. 8) y se presenta el respectivo código C# de dicho método (Figura 3. 9).

```

ALTER PROCEDURE [tarea].[Sp_GuardarCabecera]
    @titulo varchar(100),
    @objetivoProceso varchar(500),
    @fecha datetime,
    @idUsuarioAdministrador int,
    @idUsuarioSupervisor int,
    @progreso numeric(3,2)
AS
BEGIN
    DECLARE @idCabeceraTarea int
    DECLARE @tareaTarea varchar(MAX) = NULL
    DECLARE @mensaje varchar(MAX)
    DECLARE @nombreUsuario varchar(100)

    SELECT TOP 1 @idCabeceraTarea = idCabeceraTarea+1 FROM CabeceraTarea ORDER BY idCabeceraTarea DESC

    INSERT INTO CabeceraTarea(idCabeceraTarea, titulo, objetivoProceso, fechaInicio, fechaLimite, idUsuarioAdministrador, terminado, idUsuarioSupervisor,
    progreso)
    VALUES (@idCabeceraTarea, @titulo, @objetivoProceso, GETDATE(), @fecha, @idUsuarioAdministrador, 0, @idUsuarioSupervisor, @progreso)
    SELECT @nombreUsuario = nombreCompleto FROM usuario.Usuario
    INNER JOIN AspNetUsers ON AspNetUsers.Id = usuario.idIdentity
    WHERE usuario.idUsuario = @idUsuarioSupervisor
    SELECT @mensaje = 'Estimado(a) '+@nombreUsuario+' se le ha delegado la supervision de las tareas '+
    +'correspondientes al proceso titulado: '+@titulo

    EXEC dbc|Sp_GuardarMensajeSupervisor @idCabeceraTarea, @idUsuarioSupervisor, @mensaje
END

```

**Figura 3. 8.** Código SQL del Sp\_GuardarCabecera.  
Fuente: Autores

```

-----
/// <summary>
/// Para el SP que guarda la cabecera y las tareas de definicion de un proceso
/// </summary>
/// <param name="_objCabecera"></param>
1 referencia | 0 excepciones
public void GuardarCabeceraTarea(entidadCabeceraTarea _objCabecera)
{
    LogicTarea _objTarea = new LogicTarea();
    LogicMensaje _objCatalogoMensaje = new LogicMensaje();
    entidadMensaje _objMensaje = new entidadMensaje();
    LogicUsuario _objUsuario = new LogicUsuario();

    int _id_usuario = _objUsuario.ObtenerUsuarios().Where(i => i._id_identity == _objCabecera._id_identity).First()._id_usuario;

    _objConexion.Sp_GuardarCabecera(_objCabecera._titulo, _objCabecera._objetivo_proceso, _objCabecera._fecha_limite, _id_usuario,
    _objCabecera._id_usuario_supervisor, _objCabecera._progreso);

    _objTarea.GuardarTarea(_objCabecera._lista_tarea);
}

```

**Figura 3. 9.** Código C# del método GuardarCabeceraTarea.  
Fuente: Autores

Después, se crearon los respectivos controladores que permiten la comunicación sencilla y segura entre la API y las peticiones de las aplicaciones clientes. Los métodos públicos de los controladores fueron establecidos de acuerdo a las restricciones propuestas para el diseño adecuado de una API REST, codificándose los métodos HTTP: GET, POST, PUT, DELETE. A fin de continuar con el ejemplo anterior, en la Figura 3. 10 se presenta el método POST que permite que una aplicación cliente pueda crear las tareas de definición de un proceso administrativo.

```

,
/// <summary>
/// Guarda la cabecera y las tareas que se necesitan para crear un proceso
/// </summary>
/// <param name="_objCabecera"></param>
/// POST: api/Cabecera
0 referencias | 0 solicitudes | 0 excepciones
public void Post([FromBody]entidadCabeceraTarea _objCabecera)
{
    LogicCabeceraTarea _objCabeceraTarea = new LogicCabeceraTarea();
    try
    {
        _objCabeceraTarea.GuardarCabeceraTarea(_objCabecera);
    }
    catch (Exception)
    {
    }
}
}

```

**Figura 3. 10.** Código C# para el método POST para guardar una instancia.

**Fuente:** Autores

Es decir, que para completar todos los requerimientos funcionales establecidos en la etapa de planificación, en primer lugar se crearon los SP, a continuación las correspondientes clases en las carpetas “Entidades” y “Logic”, para que finalmente se establezcan los métodos HTTP en los respectivos controladores. Todo esto fue realizado mediante la ejecución de los incrementos planificados y el posterior control y elaboración de los informes de los avances alcanzados por cada incremento (Anexo 5).

#### **4. Programación de pantallas, filtros y enlaces activos:**

Para esta tarea intervinieron dos partes: primero, dado que el proyecto trata sobre el desarrollo del núcleo o motor de un Workflow, no se diseñaron las interfaces clientes, y como se ha venido mencionando, este trabajo correspondió a una tesis complementaria. Segundo, la interfaz para que los clientes puedan consultar y consumir los recursos web brindados por la API, es presentada mediante la herramienta Swagger UI, misma que permitió crear un entorno sencillo y amigable. Para lo cual, durante la etapa de desarrollo de los recursos web, fue necesario documentar todos los métodos HTTP y las variables de cada entidad, permitiendo que los clientes obtengan una descripción clara de tales elementos.

POST /api/Cabecera Guarda la cabecera y las tareas que se necesitan para crear un proceso.

Response Class (Status 200)  
OK

Model Example Value

```
[
  {
    "_objUsuario": {
      "_lista_caracteristica": [
        {
          "_id_usuario_caracteristica": "string",
          "_registro": "string",
          "_objCaracteristica": {
            "_id_caracteristica": "string",
            "_campo": "string",
            "_objTipo": {
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
_objCabecera	(required)	Datos del proceso, incluida la lista de tareas	body	Model Example Value

Parameter content type: application/json

```
"_descripcion": "string",
  "_estado": "string",
  "_fecha_envio": "2017-09-12T15:44:25.292Z",
  "_fecha_entrega": "2017-09-12T15:44:25.292Z",
  "_fecha_limite": "2017-09-12T15:44:25.292Z",
  "_tipo_tarea": "string",
  "_objUsuario": {
    "_lista_caracteristica": [
      {
        "_id_usuario_caracteristica": "string",
        "_registro": "string"
```

Figura 3. 11.Vista de swagger en la ejecución de un método.

Fuente: Autores

## 5. Programación de notificaciones, escalaciones y reportes:

Las notificaciones a los usuarios son retornadas por el núcleo, para que las aplicaciones cliente (interfaces) las muestren de acuerdo a sus necesidades, se determinan dos tipos de mensajes que serán retornados: las notificaciones nuevas (aquellas que el usuario no ha visto) y las alertas.

Las escalaciones serán realizadas mediante la verificación periódica de las tareas (diseño) y tramites (ejecución) de un proceso, realizada por los Jobs creados en la herramienta SQL Server Agent, además interviene el servicio de ejecución, el cual establece cuales son los trámites a realizar de acuerdo al flujo establecido previamente. El supervisor actúa como unidad superior en el caso de reasignación de usuario de los trámites y al tomar la decisión que crea conveniente al ocurrir un error durante la ejecución de la instancia.

El sistema permite que los usuarios (Administrador y Supervisor) pueden generar diversos tipos de reportes cuyos datos serán consultados a la BD y entregados a las interfaces. De estos documentos quedará un registro en la tabla reportes de la BD, con el fin de que el administrador conozca sobre el mismo y pueda generarlo en caso de ser necesario.

#### **3.1.2.4. PRUEBAS DEL SISTEMA**

Para alcanzar esta etapa, se comprobó que los procesos y actividades del sistema cumplen con las exigencias del proyecto. A diferencia de lo que recomienda la metodología, no se pudieron establecer previamente reglas de negocio y procesos estáticos, ya que el núcleo permite la creación dinámica de procesos, aplicar reingeniería de los mismos y creación de constantes de nuevas reglas, por lo cual fue necesario aplicar pruebas a dos procesos administrativos de la ESPAM MFL que fueron instanciados en el ambiente de producción de la herramienta Workflow, a fin de comprobar el correcto funcionamiento del núcleo. Pero esto se realizó luego de la respectiva instalación del sistema. Por lo cual, para determinar que el núcleo estaba listo para ser ejecutado en los servidores de la institución, fue necesario aplicar pruebas locales que permitieran comprobar que el sistema cumplía con los requerimientos funcionales del proyecto.

La gestión de pruebas se realizó mediante un Plan de pruebas (Anexo 6), donde se aplicaron cuatro etapas para la verificación del correcto funcionamiento del motor, estas etapas se detallan a continuación:

- 1. Pruebas Unitarias:** se verificó el correcto funcionamiento de los 27 Requerimientos Funcionales (Cuadro 4. 13) del núcleo, tomando como punto de partida los resultados que se esperaban de cada prueba que permitieron realizar el contraste respectivo con los resultados obtenidos.
- 2. Pruebas de Regresión:** se llevó a cabo la retroalimentación o verificación de los recursos que presentaron errores durante las pruebas unitarias.

**3. Pruebas Funcionales:** Estas pruebas se realizaron luego de que el núcleo fue integrado con las interfaces, desarrolladas en la tesis complementaria, para verificar que los recursos se acoplen a las peticiones de las aplicaciones cliente. Para lo cual, se enviaron diferentes entradas para verificar que el sistema realiza las transacciones y cálculos de forma correcta, devolviendo las salidas o respuestas esperadas, por ejemplo: el cálculo automático de las fechas límites para cada trámite o tarea del proceso instanciado.

Las pruebas fueron ejecutadas bajo condiciones controladas, a fin de verificar que se comporta como se esperaba, para detectar posibles errores y ejecutar la correspondiente gestión de incidencias que permita corregir esas falencias.

Otro factor importante que se consideró durante la ejecución de pruebas fue el tiempo de respuesta por parte de cada recurso web, a fin de comprobar que no solo se construyó un sistema robusto, sino también ágil y eficiente.

#### **3.1.2.5. DOCUMENTOS FORMALES DEL SISTEMA**

Luego de que se concluyeron, de forma exitosa, todas las pruebas que contribuyeron a que el núcleo sea un sistema de calidad, mediante la verificación exhaustiva del correcto cumplimiento de todos los requerimientos funcionales definidos en el Anexo 2, se procedió con la documentación formal del sistema. Aunque la metodología sugiere la elaboración de un Manual de Usuario, no fue necesario para este trabajo de titulación, dado que se trata del desarrollo de una API Web, la cual va a ser únicamente consumida y mantenida por personas expertas en el desarrollo web del lado del servidor (back-end).

Por tal motivo, siendo necesario facilitar el entendimiento del sistema para que pueda ser utilizado y posteriormente retroalimentado de forma eficiente por los integrantes de la UPS, se realizó la documentación de todos los métodos y procedimientos que conforman el sistema. La programación del núcleo fue

desarrollada a la par de los correspondientes incrementos planificatorios detallados de forma independiente en el Anexo 5, donde se incluye el Incremento 013 que especifica la elaboración del Manual de Programador (Anexo 7).

El documento incluye las especificaciones de hardware y software, el análisis y diseño del sistema y detalla de forma precisa las entidades con sus respectivas variables, los catálogos con sus métodos y los controladores que conforman los diferentes recursos brindados por el núcleo del Workflow. Además, se incluyen los diagramas “entidad-relación” pertenecientes a los respectivos esquemas de la Base de Datos del sistema, con el correspondiente diccionario de datos, generado de forma automática mediante la herramienta Data Dictionary Creator (DDC).

El Manual fue escrito siguiendo las directrices brindadas por la UPS, según el Instructivo para realizar el Manual Técnico del Programador con código INS-MTP-001, donde se especifican todos los contenidos necesarios para elaborar el documento.

### **3.1.2.6. IMPLEMENTACIÓN DEL SISTEMA**

Una vez finalizadas las etapas de análisis, diseño, programación, evaluación y documentación del núcleo del Workflow, se llevó a cabo la entrega formal e implementación del sistema en los servidores de la ESPAM MFL, mediante la realización de las siguientes actividades:

- **Configuración del Servidor de Producción y de la Base de Datos:** dado que el núcleo del Workflow y su BD fueron elaborados para ser implementados en las instalaciones de la ESPAM MFL, no fue necesario realizar la respectiva instalación de los servidores, ya que éstos tienen mucho tiempo en etapa de producción. Además, durante la fase inicial de este trabajo, se llevó a cabo el análisis de las herramientas para el desarrollo y entre otros parámetros, se consideró la plataforma sobre la que

iba a ser publicado el sistema, lo cual permitió que no existieran problemas de compatibilidad entre la arquitectura de desarrollo y la de producción.

- **Migración del Sistema:** Antes de la publicación del sistema, los integrantes manejaban la programación desde sus equipos, por lo tanto, en esta etapa se llevó a cabo la entrega formal mediante dispositivos de almacenamiento externo a los integrantes de la UPS para su la publicación del sistema en el IDE Visual Studio 2015 y la BD en el SGBD Sql Server 2008 R2, implementados en los servidores de la ESPAM MFL.
- **Pruebas del Sistema en el Ambiente de Producción:** luego de que el sistema estuvo publicado, se verificó la conexión con la BD. Posteriormente, se llevó a cabo el diseño e implementación de varios procesos de prueba, a fin de verificar el correcto funcionamiento del sistema.

# CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

## 4.1. RESULTADOS

### 4.1.1. REALIZAR EL PLAN DEL PROYECTO, IDENTIFICANDO LOS REQUERIMIENTOS.

#### 4.1.1.1. PLAN DE PROYECTO

Permitió identificar los requerimientos generales del software, las herramientas de desarrollo, los objetivos y estructura del proyecto, los entregables que serán generados, entre otros aspectos. Por lo cual, este Plan de Desarrollo de Software es a su vez uno de los entregables de MEIDAW y provee una visión global del enfoque de desarrollo propuesto. Los principales usuarios interesados en este documento son presentados en el Cuadro 4. 1.

**Cuadro 4. 1.** Usuarios interesados en el Plan de Proyecto.

<b>Tipo</b>	<b>Jefe del proyecto</b>	<b>Miembros del equipo de desarrollo</b>
<b>Involucrados</b>	Ing. Ramón Joffre Moreira Pico	Dayana Helen Bailón Delgado José Simón Mera Cantos
<b>Descripción</b>	Lo utilizó para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.	Lo utilizaron para entender lo que tenían que hacer, cuándo tuvieron que hacerlo y cuáles otras actividades dependieron de ello.

**Fuente:** Autores

Es importante mencionar que en este apartado se presenta una síntesis del Plan de proyecto, dado que el documento completo está detallado en el Anexo 1. En el Cuadro 4. 2 se especifican las secciones de la estructura completa del mismo.

**Cuadro 4. 2.** Resumen del Plan de Proyecto.

<b>Introducción</b>	Especifica propósito, alcance y resumen del documento.
<b>Vista General del Proyecto</b>	Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los entregables que serán producidos y utilizados.
<b>Organización del Proyecto</b>	Describe la estructura organizacional del equipo de desarrollo.
<b>Gestión del Proceso</b>	Explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.
<b>Especificaciones del Producto</b>	Proporciona una vista global del funcionamiento del producto y determina las herramientas que serán utilizadas.
<b>Glosario y Abreviaciones</b>	Especifica la definición de los términos y abreviaciones utilizadas.
<b>Referencias</b>	Detalla las fuentes bibliográficas utilizadas.

Fuente: Autores

### • IDENTIFICACIÓN DE REQUERIMIENTOS GENERALES

Es necesario que el núcleo brinde soporte a las interfaces definidas por la WfMC, a fin de que pueda proveer el control del ambiente de ejecución de las instancias de los flujos de trabajo. Es por esto, que se utilizaron dichas interfaces como referencia, detallándose en el Cuadro 4. 3 los requerimientos generales que debe alcanzar, a través de la realización o cumplimiento de los requerimientos funcionales y no funcionales descritos en el Cuadro 4. 13.

**Cuadro 4. 3.** Requerimientos generales del núcleo.

<b>Identificación</b>	<b>Nombre</b>	<b>Característica</b>
<b>RG01</b>	Herramientas de definición de procesos.	Para el manejo de la gestión de los procesos administrativos y sus respectivas versiones.
<b>RG02</b>	Aplicaciones clientes	Para la ejecución de las tareas y actividades de los flujos de trabajo.
<b>RG03</b>	Aplicaciones Invocadas	Para levantar aplicaciones de terceros, que contribuyan en el desarrollo del flujo de trabajo.
<b>RG04</b>	Herramientas de administración y monitoreo.	Para verificar el cumplimiento de las tareas de los flujos de trabajo.

Fuente: Autores.

### • DEFINIR EL ESQUEMA DE FUNCIONAMIENTO DEL NÚCLEO

Especifica la comunicación entre el motor y los demás elementos que intervienen en el Workflow; las peticiones del usuario mediante las interfaces, las consultas de datos a la BD interna de la herramienta y a las externas (institucionales); y las interacciones con aplicaciones de terceros (Figura 4. 1).

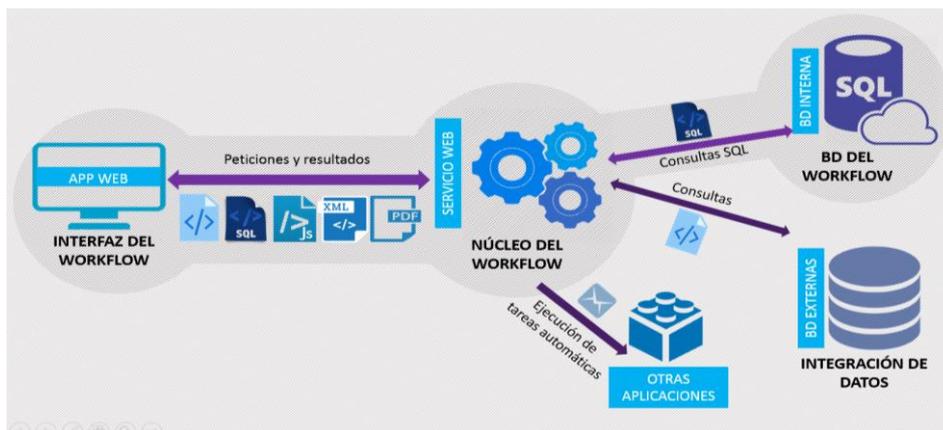


Figura 4. 1. Esquema general del funcionamiento del núcleo.

Fuente: Autores

- **DETERMINAR LAS HERRAMIENTAS DE DESARROLLO**

En el Cuadro 4. 4 se presenta una tabla con las principales herramientas tecnológicas seleccionadas por los autores, que permitieron el desarrollo del núcleo del Workflow de acuerdo a lo planificado.

Cuadro 4. 4. Herramientas y lenguaje de programación para el desarrollo del núcleo.

Lenguaje de programación	Entorno de Desarrollo Integrado (IDE)	Sistema Gestor de Base de Datos (SGBD)	Framework
Visual C#	Visual Studio (VS)	SQL Server 2008 R2 Express	ASP.NET WebAPI

Fuente: Autores

- **ESTRUCTURA DEL PROYECTO**

En el Cuadro 4. 5 se presentan y describen todos los documentos, denominados también entregables, que fueron elaborados y utilizados durante el desarrollo de este trabajo.

**Cuadro 4. 5.** Lista de entregables.

<b>Número</b>	<b>Entregable</b>	<b>Descripción</b>
1	Plan de Desarrollo de Software	Provee una visión global del enfoque de desarrollo propuesto.
2	Especificación de Requisitos de Software (ERS)	Los requisitos del núcleo, basándose en las directrices dadas por el estándar Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.
3	Diagramas UML: Casos de Uso.	Desde la perspectiva de actores y sistemas externos, detalla el comportamiento del núcleo mediante los diagramas UML.
4	BPMN 2.0.	Contiene los elementos del estándar BPMN 2.0 utilizados por el modelador, además especifica sus características, definición y los atributos utilizados y las posibles formas que pueden conectarse entre sí.
5	Plan de iteración	Un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.
6	Plan de pruebas	Especifica las condiciones de los dos casos de prueba: entradas, condiciones de ejecución y resultados esperados. Al ejecutar las pruebas, se presentaran los resultados obtenidos y se compararán con los resultados esperados. Además, cada caso de prueba lleva asociado un procedimiento de prueba con las instrucciones para realizar la misma.
7	Manual del programador	De forma clara y precisa se detalla la documentación del código, principalmente las clases, objetos, variables, métodos y procedimientos utilizados durante la programación, además del diccionario de datos correspondiente a la BD del sistema.
8	Producto	Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la Fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva versión al final de cada iteración.

**Fuente:** Autores

### • CRONOGRAMA DE DESARROLLO DEL PROYECTO

Se consideró como resultado principal de este documento, el cronograma de desarrollo presentado en el Cuadro 4. 6, donde se especifican las sub-actividades, responsables y tiempos necesarios durante el desarrollo del núcleo.

**Cuadro 4. 6.** Cronograma de desarrollo del proyecto.

Actividad	Sub-actividad	Responsables	Duración (días)	Fecha de inicio	Fecha de fin
Configurar la arquitectura.	Instalar Visual Studio 2015.	Dayana Bailón	2	05/12/16	06/12/16
	Instalar SQL SERVER 2008.	José Mera	3	07/12/16	09/12/16
Diseñar e implementar la arquitectura de datos.	Diseñar el esquema de la BD.	Dayana Bailón José Mera	10	12/12/16	23/12/16
	Implementar el esquema en SQL SERVER 2008.	Dayana Bailón José Mera	5	26/12/16	30/12/16
	Desarrollar Stores Procerdures	Dayana Bailón José Mera	5	02/01/17	06/01/17
Desarrollar la capa intermedia (Negocio).	Crear capa de acceso a la BD.	José Mera	1	09/01/17	09/01/17
	Desarrollar métodos CRUD.	Dayana Bailón José Mera	144	10/07/17	28/07/17
	Desarrollar Servicios Web.	Dayana Bailón José Mera	144	10/07/17	28/07/17
Programar notificaciones y escalaciones	Configurar un cron job para verificar el cumplimiento de las tareas.	Dayana Bailón José Mera	20	03/07/17	28/07/17
	Solicitar la documentación de un proceso administrativo.	José Mera	1	01/08/17	02/08/17
Diseño del plan de pruebas	Establecer un proceso, como sujeto de prueba.	Dayana Bailón	1	02/08/17	02/08/17
	Realizar una entrevista al Coordinador de la UPS para conocer los detalles del proceso seleccionado.	Dayana Bailón José Mera	1	03/08/17	03/08/17
	Realizar la documentación del plan de pruebas.	Dayana Bailón José Mera	2	03/08/17	04/08/17
	Ejecución de pruebas funcionales.	José Mera	5	07/08/17	11/08/17
Ejecución de pruebas	Documentar los errores.	Dayana Bailón	5	07/08/17	11/08/17
	Ejecución de prueba de carga de datos.	José Mera	5	14/08/17	18/08/17
	Documentar los errores.	Dayana Bailón	5	14/08/17	18/08/17
Gestión de incidencias	Corregir los errores documentados.	Dayana Bailón José Mera	5	14/08/17	18/08/17
	Realizar el manual del programador y del administrador.	Dayana Bailón José Mera	9	21/08/17	31/08/17
Integrar el Núcleo con las interfaces	Permitir acceso a los recursos.	Dayana Bailón	5	04/09/17	08/09/17
Instalar el sistema en el servidor	Instalar el sistema.	José Mera	5	11/09/17	15/09/17
	Verificar el correcto funcionamiento del sistema.	Dayana Bailón	10	18/09/17	29/09/17

Fuente: Autores

#### 4.1.1.2. DOCUMENTO DE ALCANCE Y REQUERIMIENTOS

Este documento es una Especificación de Requisitos Software (ERS), basado en las directrices dadas por el estándar IEEE 830, 1998. Detalla las restricciones, suposiciones, dependencias y requerimientos funcionales (RF) y no funcionales del núcleo del Workflow. Constituye el documento principal de la

primera etapa de la segunda fase de MEIDAW y está dirigido a los encargados de brindar el mantenimiento necesario del mismo.

En el Cuadro 4. 7 se describen los roles de los integrantes del proyecto, durante el desarrollo del núcleo.

**Cuadro 4. 7.** Personal involucrado en el desarrollo del proyecto.

<b>Nombre</b>	Dayana Bailón	José Mera
<b>Rol</b>	Analista, DBA (Administradora de Base de Datos) y programadora.	Analista, DBA (Administrador de Base de Datos) y programador.
<b>Responsabilidad</b>	Análisis de información, diseño y programación del núcleo del Workflow.	Análisis de información, diseño y programación del núcleo del Workflow.
<b>Información de contacto</b>	dayu_espam@outlook.es	josmer1202@hotmail.com

Fuente: Autores

Este documento consta de tres secciones, descritos en el Cuadro 4. 8. La estructura completa está ubicada en el Anexo 2, del cual se han seleccionado los ítems más importantes para ubicarlos en este apartado.

**Cuadro 4. 8.** Resumen del documento de Alcance y Requerimientos.

<b>Introducción</b>	Introducción, propósito, alcance, personal involucrado, acrónimos y definiciones, referencia y resumen del documento.
<b>Descripción General</b>	Descripción general del núcleo, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.
<b>Requisitos Específicos</b>	Se definen detalladamente los requisitos funcionales y no funcionales que debe satisfacer el desarrollo del núcleo.

Fuente: Autores

## • DESCRIPCIÓN GENERAL

El núcleo del Workflow fue desarrollado de manera independiente, con Base de Datos propia y ejecutado por un servicio REST usando ASP.NET Web API.

**CARACTERÍSTICAS DE LOS USUARIOS:** se han definido dos categorías de usuarios:

- Cargos definidos por el sistema: el Administrador, entre otras cosas, podrá especificar quienes cumplirán los siguientes roles dentro de cada proceso: Supervisor, Modelador de diagramas, Programador y DBA (Administrador

de Base de Datos). Las funciones que cumplen cada uno de ellos son detallados en el Cuadro 4. 9.

- Cargos definidos por el Administrador: dentro de los cargos definidos por el sistema, existe un rol denominado Operador, en el cual el Administrador podrá definir los cargos de las personas y/o unidades responsables de las diferentes actividades que conforman los procesos administrativos de la ESPAM MFL. Por ejemplo, secretaria.

**Cuadro 4. 9.** Características de los usuarios.

Tipo de usuario	Descripción
Administrador	Puede aprobar o no un nuevo proceso, revisar el cumplimiento de todos los trámites que intervienen en el proceso que administra, reasignar responsables y realizar reportes de dichos procesos.
Supervisor	Puede verificar el cumplimiento de todos los trámites del proceso, reasignar responsables en las actividades y realizar reportes de los procesos.
Operador	Realiza únicamente los trámites (tareas) a los que ha sido asignado. En caso de que en la etapa de diseño del proceso, durante la asignación de recursos, haya sido definido como usuario responsable del inicio del proceso, puede instanciarlo cuando sea necesario.
Modelador	Se encarga de modelar el diagrama (flujo) del proceso.
Diseñador	Es el responsable de diseñar los formularios que posteriormente permitirán que el Operador registre la respectiva información, además puede generar Vistas de datos y Reportes en PDF que presenten información relevante de los formularios, añadiéndole diseño personalizado.
DBA (Administrador de Base de Datos)	Puede realizar integraciones externas mediante la ejecución de servicios web, o gestionar integraciones internas para consultar información propia del sistema.

Fuente: Autores.

Las restricciones, suposiciones y dependencias del núcleo están ubicados en el Cuadro 4. 10.

**Cuadro 4. 10.** Restricciones, suposiciones y dependencias del núcleo.

Restricciones	Suposiciones y dependencias
Lenguajes y tecnologías en uso: ASP.NET, C#, VISUAL STUDIO 2015.	Se asume que los requisitos aquí descritos son estables.
Los servidores deben ser capaces de atender consultas concurrentemente.	Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados, para garantizar una ejecución correcta y la disponibilidad del mismo.
El sistema se diseñó según un modelo cliente/servidor.	

Fuente: Autores.

## • REQUISITOS ESPECÍFICOS

**REQUERIMIENTOS FUNCIONALES:** Los procesos administrativos de la ESPAM MFL están formados por un conjunto de actividades, ordenadas de

forma lógica, a fin de alcanzar algún objetivo previamente establecido. El núcleo es el encargado de gestionarlas, incluidos otros elementos, especificados en el Cuadro 4. 11.

**Cuadro 4. 11.** Conceptos de los elementos de los procesos administrativos.

<b>Actividades</b>	Representan la lógica para una tarea específica en un proceso de flujo de trabajo.
<b>Participantes de la actividad</b>	Usuarios que se hayan asignado para interactuar con las actividades de un proceso de flujo de trabajo, como aprobaciones, correo, solicitudes de información, etc.
<b>Escalaciones</b>	Son las vías de acceso que conectan varias actividades de flujo de trabajo, representan un flujo (continuidad), unidas en un grafo que brinde las relaciones de precedencia del proceso y las transiciones posibles entre actividades.
<b>Parámetros de entrada/salida</b>	Define los datos que se transfieren al proceso de flujo de trabajo y que se devuelven de este. En cada tarea de un proceso se debe tener en cuenta dichos parámetros.
<b>Roles propios del proceso</b>	Aquellos roles que deben ser nominados en tiempo de inicio o de ejecución del workflow y que no pueden asignarse de ante mano a un miembro o grupo de la organización.
<b>Información no estructurada</b>	El manejo de documentos adjuntos.
<b>Reglas</b>	Conjunto de reglas de negocio asociadas a los procesos, que no pueden representarse en el grafo debido a su naturaleza o a que tienen un alcance global y no particular de cada actividad. Por lo tanto, son incluidos el concepto de validaciones normales y avanzadas.
<b>Subprocesos</b>	La WfMC considera al paquete como un contenedor de diseños de procesos, que a su vez puede contener otros paquetes y además se ha dado al sistema la capacidad de definir datos relevantes para la toma de decisiones del proceso, roles, validaciones y eventos a nivel del paquete, aplicando a todos los procesos que se encuentran por debajo de este.

Fuente: Autores

Además, existen elementos que no conforman a los procesos como tal, pero que su utilización contribuirá a la calidad de la gestión administrativa de la ESPAM MFL. Tales elementos son presentados en el Cuadro 4. 12.

**Cuadro 4. 12.** Conceptos de otros elementos de los procesos administrativos.

<b>Control</b>	Sobre la evolución de cada una de las actividades de un proceso, en caso de que no se completen a tiempo por parte de los usuarios, se presentarán notificaciones a los responsables de cada tarea y al Supervisor del proceso.
<b>Reportes</b>	Facilita la toma de decisiones a los Supervisores y Administradores de los diferentes procesos, el núcleo es capaz de procesar los datos necesarios para los reportes.
<b>Integraciones</b>	Es necesario integrarse a los otros sistemas de la ESPAM MFL, por lo cual están las siguientes integraciones: de datos, funcional y documental, a fin de que el workflow no quede aislado y, al contrario, pueda ser aprovechado al máximo por la organización.

Fuente: Autores

A continuación, en el Cuadro 4. 13 se describe un resumen de los RF que contendrán las descripciones específicas del comportamiento básico del núcleo.

**Cuadro 4. 13.** Resumen de los requisitos funcionales del núcleo.

<b>Identificación</b>	<b>Nombre</b>	<b>Características</b>
RF01	Definición de tareas para la creación de un proceso.	Envío de tareas (formularios, validaciones, diagrama, etc.) necesarias para la definición de un proceso.
RF02	Supervisión de tareas para diseño del proceso.	El usuario Supervisor podrá inspeccionar el estado de las tareas de los diversos procesos que estén a su cargo.
RF03	Definición del Flujo (diagrama)	Diseño del diagrama del proceso.
RF04	Diseñador de Formularios.	Se permite al usuario Diseñador crear los formularios que se requieran para la ejecución de un proceso.
RF05	Integración Interna (II)	Se permite al DBA consultar información ingresada en el sistema por los Operadores mediante los Formularios, durante la ejecución del proceso.
RF06	Integración Externa (IE)	Se permite al DBA consultar información brindada por servicios web externos al Workflow.
RF07	Vista de datos	Se permite al usuario Diseñador generar vistas a partir de los datos obtenidos en una integración, ya sea interna o externa.
RF08	Reporte de formularios	Se permite al usuario Diseñador generar reportes en formato PDF a partir de los datos obtenidos en una integración, ya sea interna o externa.
RF09	Aprobar o rechazar tareas durante el diseño del proceso.	El usuario Administrador podrá aprobar o rechazar las tareas enviadas por los demás usuarios (Diseñador, Modelador, DBA).
RF10	Definición de un proceso.	Registro de un nuevo proceso en la BD.
RF11	Asignación de recursos al diagrama.	Se asignarán los recursos (responsables, formularios, vista de datos, plazo estimado, etc.) a las actividades, compuertas y eventos del diagrama realizado y aprobado previamente.
RF12	Instancia de un proceso.	Los usuarios operadores podrán ejecutar (instanciar) los procesos a los que hayan sido asignados.
RF13	Eliminar instancia de un proceso.	Los usuarios Supervisores podrán eliminar la instancia de un proceso, cuando haya ocurrido un error fatal.
RF14	Reiniciar instancia de un proceso.	Los usuarios Supervisores podrán reiniciar la instancia de un proceso, cuando haya ocurrido un error fatal.
RF15	Historial de instancia de un proceso.	El core llevará el registro de todos los cambios de estado de una instancia.
RF16	Rediseño de un proceso.	Los Administradores podrán hacer cambios al diseño o flujo de un proceso ya creado, asignar nuevos atributos, etc.
RF17	Supervisión de tareas (trámite) del proceso.	El usuario Supervisor podrá inspeccionar el estado de las tareas de los diversos procesos que estén a su cargo.
RF18	Ejecución de las actividades.	Las ejecuciones de las actividades serán realizadas de acuerdo al flujo establecido en el diseño del proceso instanciado.
RF19	Reiniciar actividad.	El Supervisor podrá reiniciar una actividad, en caso de que se haya excedido el tiempo límite y la actividad no haya sido finalizada.
RF20	Gestión de cargos administrativos.	Crear, modificar y eliminar los cargos administrativos.
RF21	Definición de participantes.	Determinar los responsables de las diferentes acciones en una actividad.
RF22	Reasignación de participantes.	Garantizar el poder hacer cambios de responsables de las tareas o trámites del proceso en caso de ser necesario.
RF23	Mecanismos de control	Establecer mecanismos que permitan controlar la ejecución de una actividad.
RF24	Errores durante la ejecución de un proceso.	Presentar al Administrador los posibles errores que se puedan dar durante la ejecución de un proceso y las alternativas de solución.

RF25	Notificaciones	Informar de las tareas y actividades existentes en el sistema, a los respectivos usuarios.
RF26	Reportes	Garantizar que el core devuelva datos necesarios para analizar el funcionamiento del mismo.
RF27	Integración de datos	Permitir la integración de diferentes fuentes de datos.

Fuente: Autores

**REQUERIMIENTOS NO FUNCIONALES:** El desarrollo del núcleo se realizó de acuerdo a los siguientes atributos de calidad: desempeño, disponibilidad, escalabilidad, facilidad de uso e ingreso de la información, flexibilidad, instalación, mantenibilidad, seguridad.

**REQUISITOS COMUNES DE LAS INTERFACES:** Las especificaciones de las interfaces de hardware, software y de comunicación, necesarias para el correcto funcionamiento del núcleo están especificadas en el Cuadro 4. 14.

**Cuadro 4. 14.** Requisitos comunes de las interfaces.

Interfaces de hardware	Interfaces de software	Interfaces de comunicación
Servidor con las siguientes características: Servidor Intel Xeon. 2.53 GHz RAM de 6 GB. 15 MB de cache, 64 bits. Disco duro de 2 TB 7200 rpm SATA3 Windows Server 2008 R2	Sistema Operativo: Windows server 2008. Base de Datos: Sql Server 2008 IDE: Visual Studio 2015	Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible. Por ejemplo, para transferir archivos o documentos deberán utilizarse protocolos existentes (FTP u otros convenientes).

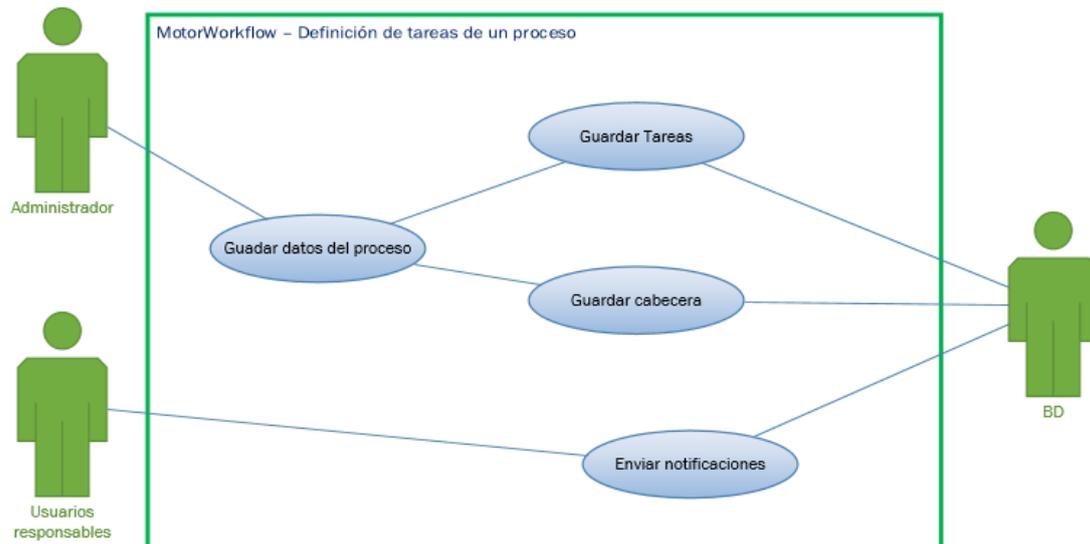
Fuente: Autores.

#### 4.1.1.3. CASOS DE USO

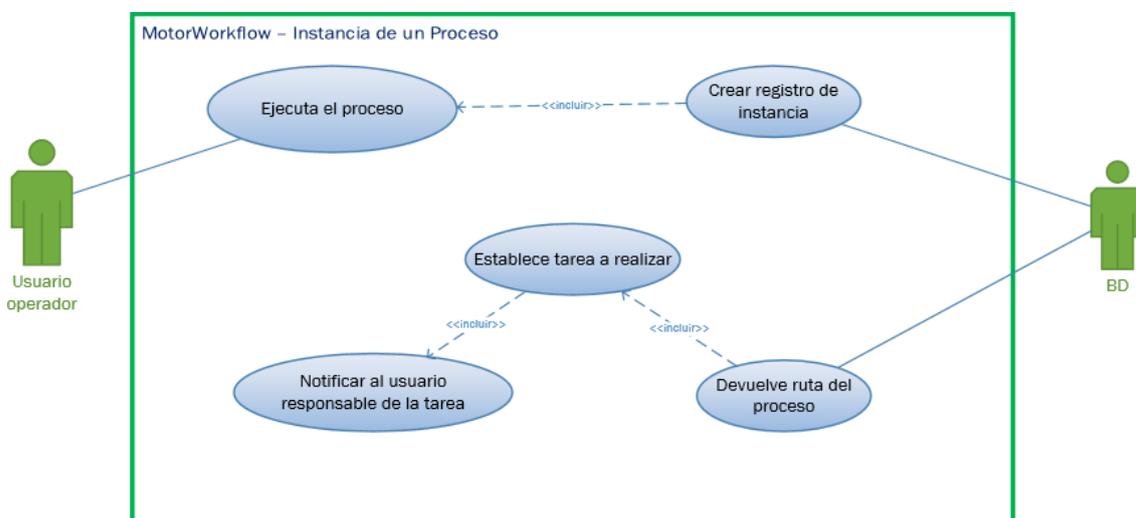
Se elaboraron 27 casos de uso, mismos que explican el comportamiento del sistema desde el punto de los clientes de la API WEB desarrollada, cada uno de estos diagramas se encuentran especificados en el anexo 3, se realizó un modelo para cada requerimiento funcional.

Para este apartado se ha considerado los diagramas UML de la Figura 4. 2, donde se detalla el comportamiento del sistema cuando se crea un nuevo proceso, es decir la parte del diseño y la Figura 4. 3 que presenta la instancia

de un proceso administrativo, es decir la ejecución del mismo. Como ya se mencionó en el párrafo anterior, estos diagramas se detallan en el anexo 3.



**Figura 4. 2.** Diagrama UML para el RF01.  
Fuente: Autores



**Figura 4. 3.** Diagrama UML del RF012.  
Fuente: Autores.

#### 4.1.1.4. INCREMENTOS PLANIFICADOS

Todos los incrementos realizados (cada 15 días) durante el desarrollo del núcleo, están detallados en el Anexo 5 de este documento. A continuación se presentan dos incrementos que corresponden a la sub-actividad: desarrollar

recursos web. Cabe mencionar que los incrementos se realizaron únicamente para la fase 2 de la metodología.

En el Cuadro 4. 15 se muestra el desarrollo de la integración interna y externa, la obtención de los datos de los formularios, las validaciones y la selección de dichos datos de acuerdo al tipo de filtro utilizado, lo cual incluye el desarrollo de procedimientos, métodos lógicos para conectar con la BD, además para la integración externa se usó una nueva clase, que permitió analizar los datos obtenidos del servicio web.

**Cuadro 4. 15.** Incremento 004.

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio:</b> 06/02/2017		<b>Fecha Fin:</b> 24/02/2017		<b>Id.</b> 004
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Desarrollo de store procedure para realizar la integración interna	Dayana Bailón	Ninguno	Ninguno
2	Desarrollo de métodos verificación y ejecución de una integración interna.	Dayana Bailón	Ninguno	Ninguno
3	Desarrollo de recurso para la integración interna	Dayana Bailón	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para realizar una integración Externa	José Mera	Ninguno	Ninguno
5	Desarrollo de métodos para la ejecución de una integración externa.	José Mera	Ninguno	Ninguno
6	Desarrollo de recursos para la verificación de la integración externa	José Mera	Notificar la realización de la URI	Ninguno
<b>EJECUCIÓN</b>				
Terminado				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS</b>	<b>POSITIVOS</b>	Implementación de nuevas librerías que permiten el análisis de un servicio web.		
	<b>NEGATIVOS</b>	Ninguno.		
<b>INFORMES DE AVANCES</b>				
<b>Estado Actual del Sistema</b>	<b>n. de actividad</b>	<b>Completada</b>	<b>Retrasada</b>	
El desarrollo del núcleo avanza, se cuenta con la validación de las integraciones internas, obteniendo los datos registrados por medio de los formularios. Además se pueden implementar los servicios web	1	✓	-	
	2	✓	-	
	3	✓	-	
	4	✓	-	
	5	✓	-	
	6	✓	-	
<b>Elaboración:</b> Autores.				

Las actividades presentadas en el Cuadro 4. 16 corresponde a la ejecución de un proceso, para lo cual el administrador debe haber realizado la asignación de recursos. Esta ejecución comprende en el inicio de una nueva instancia de un proceso, así como la ejecución de las actividades. Además se realizan actividades correspondientes a la supervisión de las instancias del proceso. En este incremento se incluye la verificación automática de las fechas límite de las actividades que no han sido realizadas como mecanismo para controlar el cumplimiento por parte de los operadores.

**Cuadro 4. 16.** Incremento 011.

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio:</b> 03/07/2017		<b>Fecha Fin:</b> 21/07/2017		<b>Id.</b> 011
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Desarrollo de store procedure para iniciar la instancia de un proceso.	José Mera	Ninguno	Ninguno
2	Desarrollo de los métodos para iniciar una instancia.	José Mera	Ninguno	Ninguno
3	Desarrollo de recurso para iniciar una instancia	José Mera	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para eliminar una instancia	Dayana Bailón	Ninguno	Ninguno
5	Desarrollo de los métodos para eliminar una instancia.	Dayana Bailón	Ninguno	Ninguno
6	Desarrollo de recurso para eliminar una instancia	Dayana Bailón	Notificar la realización de la URI	Ninguno
7	Desarrollo de store procedure para reiniciar la instancia de un proceso.	Dayana Bailón	Ninguno	Ninguno
8	Desarrollo de los métodos para reiniciar una instancia.	Dayana Bailón	Ninguno	Ninguno
9	Desarrollo de recurso para reiniciar una instancia	Dayana Bailón	Notificar la realización de la URI	Ninguno
10	Implementar autenticación OAuth 2.0a swagger.	José Mera	Ninguno	Ninguno
11	Desarrollo de store procedure para la ejecución de las actividades (trámites).	José Mera	Ninguno	Ninguno
12	Desarrollo de métodos para la ejecución de los trámites.	José Mera	Ninguno	Ninguno
13	Desarrollo de recursos para la ejecución de los trámites	José Mera	Notificar la realización de la URI	Ninguno

14	Desarrollo de store procedure para reiniciar un trámite.	Dayana Bailón	Ninguno	Ninguno
15	Desarrollo de los métodos para reiniciar un trámite.	Dayana Bailón	Ninguno	Ninguno
16	Desarrollo de recurso para reiniciar un trámite.	Dayana Bailón	Notificar la realización de la URI	Ninguno
17	Desarrollo de métodos para supervisión de trámites	José Mera	Ninguno	Ninguno
18	Desarrollo de recursos para supervisión de los trámites.	José Mera	Notificar la realización de la URI	Ninguno

**EJECUCIÓN**

Terminado

**CONTROL DE INCREMENTOS****FEEDBACKS POSITIVOS** Se logra implementar la autenticación a Swagger**NEGATIVOS** Ninguno**INFORMES DE AVANCES**

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El núcleo permite que se inicie la instancia de un proceso, además de que en caso de presentarse un error se pueda gestionar el mismo, dándole algunas opciones al supervisor del mismo. Se ejecutan las actividades, las cuales son monitoreadas por el supervisor, en caso de existir un error durante la ejecución de la misma, esta puede ser reiniciada.	1	✓	-
	2	✓	-
	3	✓	-
	4	✓	-
	5	✓	-
	6	✓	-
	7	✓	-
	8	✓	-
	9	✓	-
	10	✓	-
	11	✓	-
	12	✓	-
	13	✓	-
	14	✓	-
	15	✓	-
	16	✓	-
	17	✓	-
	18	✓	-

**Elaboración:** Autores.

## 4.1.2. ELABORAR EL NÚCLEO CON BASE A LOS REQUERIMIENTOS ESTIPULADOS

### 4.1.2.1. DIAGRAMAS ENTIDAD-RELACIÓN DE LA BASE DE DATOS DEL NÚCLEO DE WORKFLOW

- **DIAGRAMA 1:** La Figura 4. 4 presenta el diagrama con las relaciones entre las tablas que permiten crear y modificar (control de versionamiento) los procesos administrativos de la ESPAM MFL.

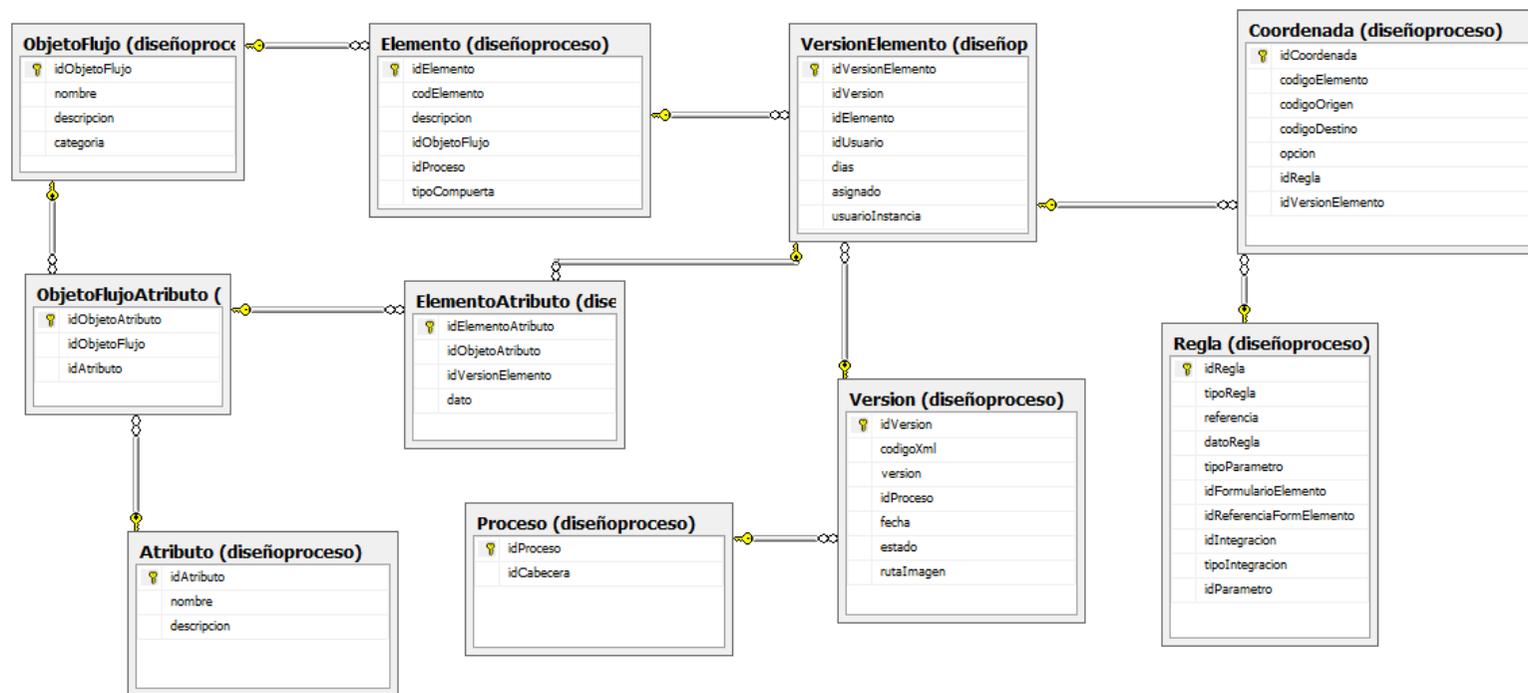


Figura 4. 4. Diagrama 1 - Gestión de los procesos administrativos, con sus respectivas versiones.

Fuente: Autores

- **DIAGRAMA 2:** En la Figura 4. 5 se presenta el diagrama con las relaciones de las tablas que permiten la ejecución (instancia) de los procesos administrativos y el almacenamiento de los datos e información, incluidos los archivos adjuntos.

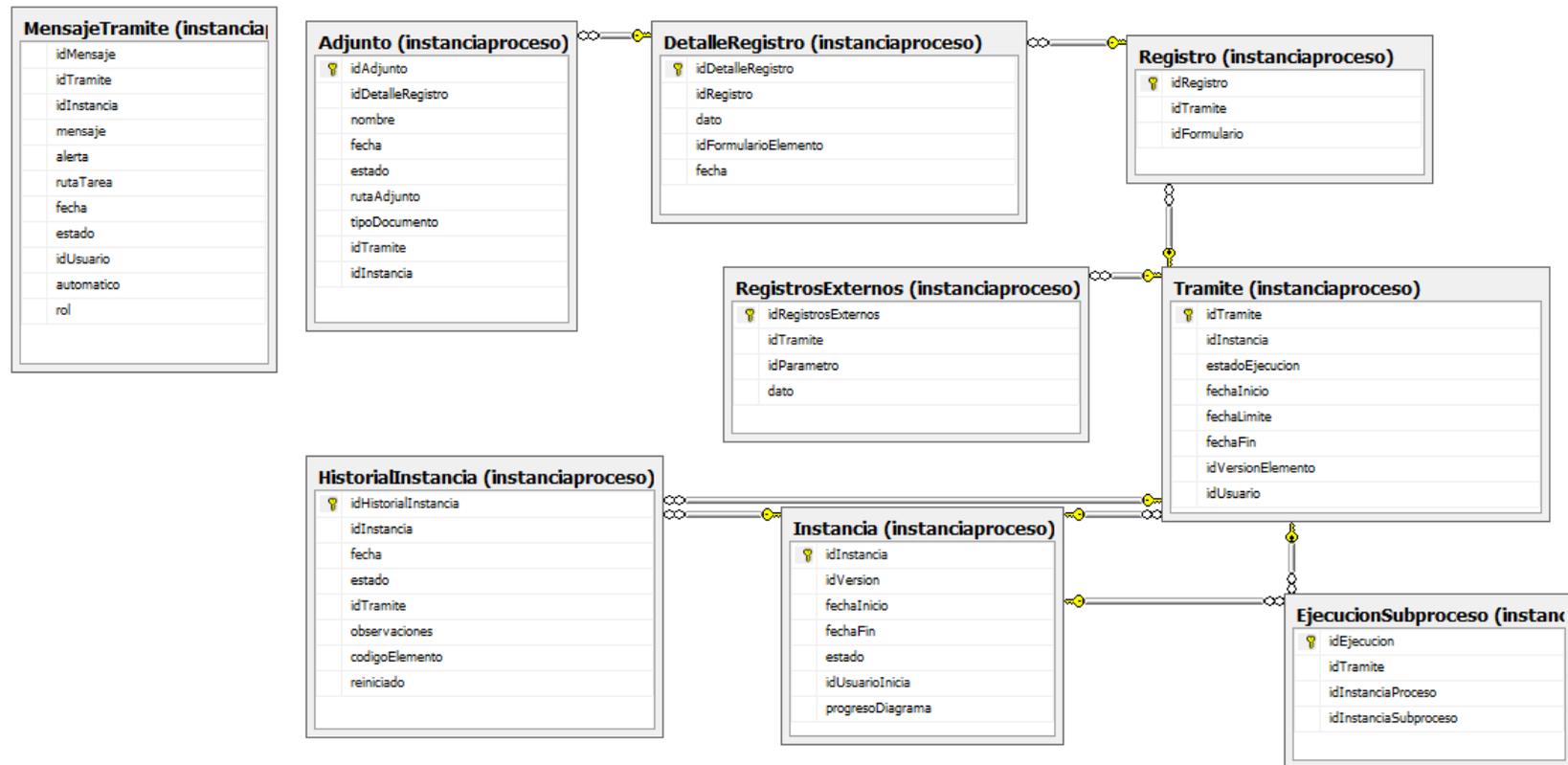


Figura 4. 5. Diagrama 2 - Ejecución (instancia) de los procesos administrativos.

Fuente: Autores.

- **DIAGRAMA 3:** En la Figura 4. 6 se presenta el diagrama con las relaciones de las tablas que permiten el login de los usuarios administrativos de la ESPAM MFL, así como la definición de sus roles y características.

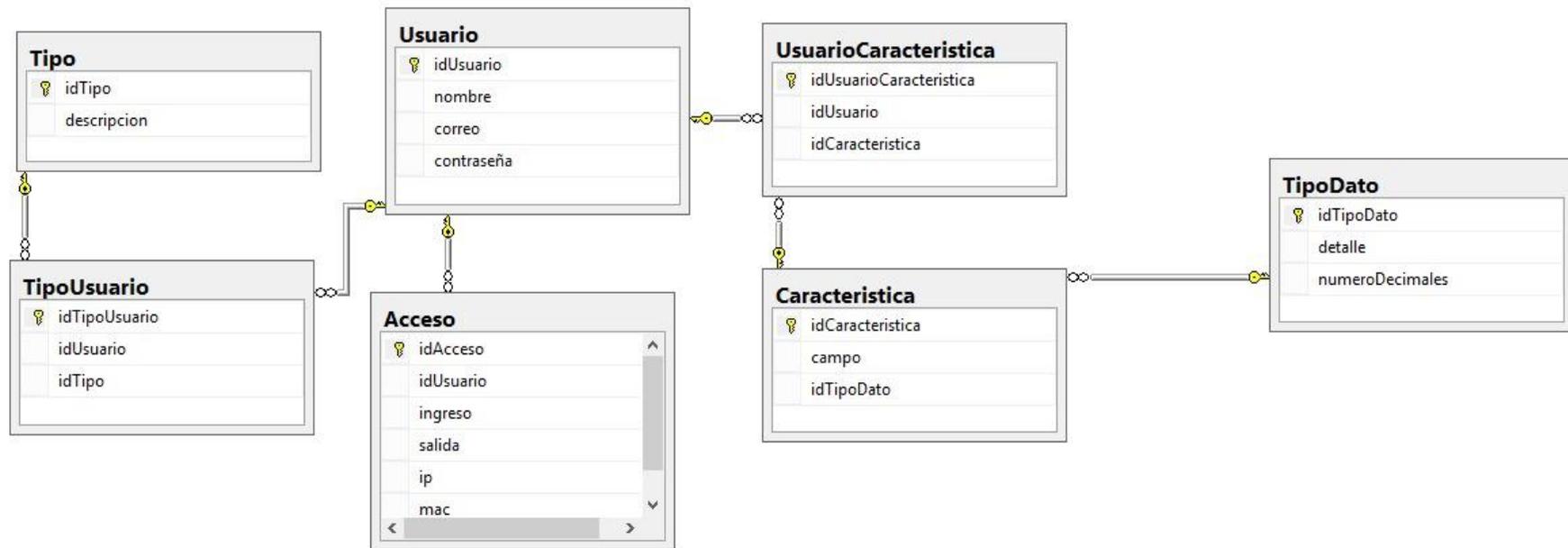
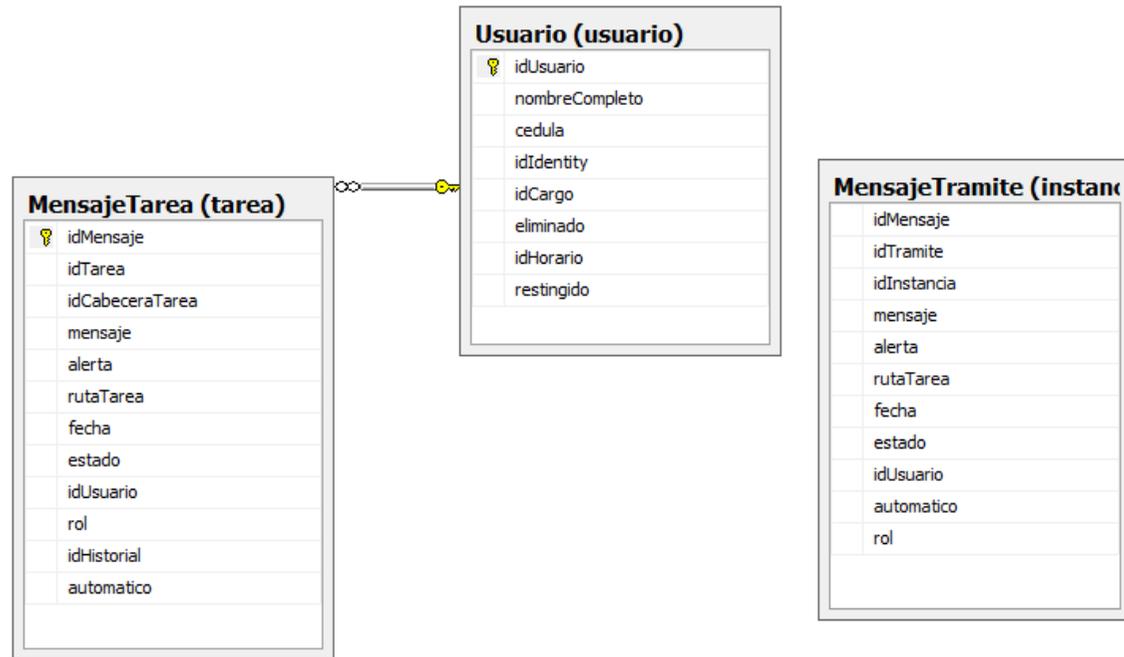


Figura 4. 6. Diagrama 3- Login de los usuarios administrativos de la ESPAM MFL, con sus roles y características.

Fuente: Autores.

- **DIAGRAMA 4:** En la Figura 4. 7 se presenta el diagrama con las relaciones de las tablas que permiten que el sistema notifique a los usuarios que intervienen en los flujos de los procesos administrativos de la ESPAM MFL. Divididos en notificaciones para las tareas durante el diseño del proceso y los trámites durante la ejecución (instancia) del proceso.



**Figura 4. 7.** Diagrama 4 - Envío y recepción de notificaciones y mensajes para los usuarios.

**Fuente:** Autores.

- **DIAGRAMA 5:** En la Figura 4. 8 se presenta el diagrama con las relaciones de las tablas que permiten establecer las tareas y responsables para armar un nuevo proceso (o versión) administrativo de la ESPAM MFL. Por cuestión de presentación no se muestran las relaciones entre las tablas completas.

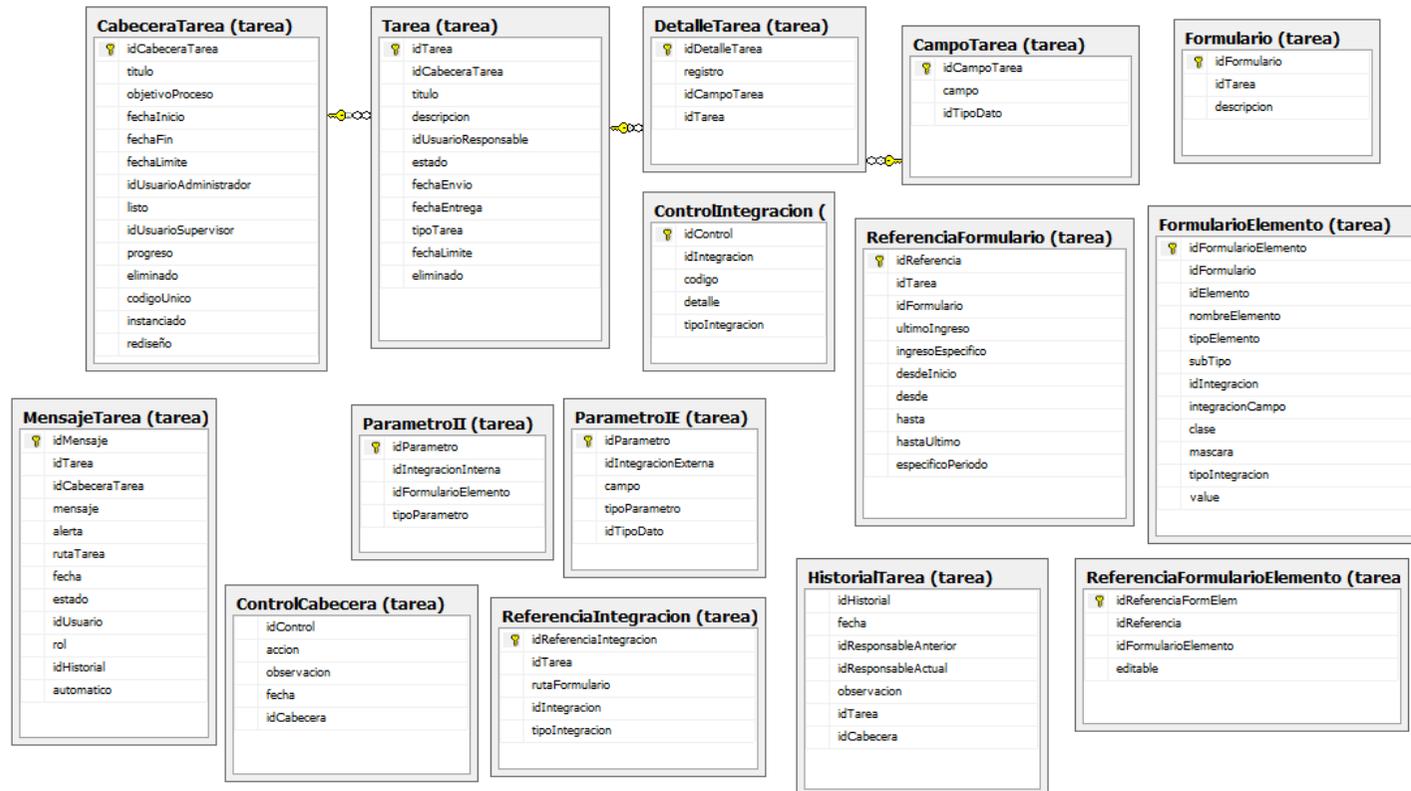


Figura 4. 8. Diagrama 5 - Establecer las tareas y responsables para armar un nuevo proceso (o versión) administrativo.

Fuente: Autores

### 4.1.2.2.STORE PROCEDURE

La Figura 4. 9 muestra el CRUD (Create, Read, Update and Delete) de una tabla como ejemplo de la manera en que se realizaron algunos procedimientos almacenados en la base de datos. Se tomó como referencia la tabla Usuarios.

<p style="text-align: center; color: #4F81BD; font-weight: bold;">CREATE</p> <pre> CREATE PROCEDURE [diseño proceso].[Sp_GuardarUsuarios]     @nombre varchar(100),     @correo varchar(50),     @contraseña varchar(50) AS BEGIN     DECLARE @idUserario int      SELECT @idUserario = COUNT(idusuario) FROM usuario.Usuario      IF @idUserario = 0         SET @idUserario = 1     ELSE         SET @idUserario = @idUserario + 1      INSERT INTO usuario.Usuario(idUsuario, nombre, correo, contraseña)     VALUES (@idUserario, @nombre, @correo, @contraseña) END </pre>	<p style="text-align: center; color: #4F81BD; font-weight: bold;">READ</p> <pre> ALTER PROCEDURE [diseño proceso].[Sp_ObtenerUsuarioPorId]     @idUserario int AS BEGIN     SELECT usuario.Usuario.nombre, usuario.Usuario.correo, usuario.Usuario.contraseña,         usuario.Usuario.idUsuario     FROM usuario.Usuario     WHERE usuario.Usuario.idUsuario= @idUserario END </pre>
<p style="text-align: center; color: #4F81BD; font-weight: bold;">UPDATE</p> <pre> CREATE PROCEDURE [diseño proceso].[Sp_ActualizarUsuario]     @idUserario int,     @nombre varchar(100),     @correo varchar(50),     @contraseña varchar(50) AS BEGIN     UPDATE usuario.Usuario SET nombre=@nombre, correo=@correo, contraseña=@contraseña     WHERE idUsuario=@idUserario END </pre>	<p style="text-align: center; color: #4F81BD; font-weight: bold;">DELETE</p> <pre> CREATE PROCEDURE [diseño proceso].[Sp_EliminarUsuario]     @idUserario int AS BEGIN     DELETE FROM usuario.Usuario     WHERE idUsuario=@idUserario END </pre>

**Figura 4. 9.** Storage procedure para la tabla usuarios  
Fuente: Autores

Es importante mencionar que también existieron entidades que no solo requerían especificar el CRUD, sino establecer otro conjunto de sentencias, pero dado que son muchos SP, en el Cuadro 4. 17 se presenta la descripción o funcionalidad de los SP más importantes de la base de datos del sistema, categorizados por esquemas y de acuerdo a los RF que pertenecen. Por su parte, el Cuadro 4. 18 presenta los SP para las notificaciones y reportes y el Cuadro 4. 19 los SP como herramienta o mecanismo de control.

**Cuadro 4. 17.** Procedimientos almacenados del sistema agrupados por esquema.

Esquema	Requerimiento Funcional	Procedimientos almacenados que intervienen	Descripción del procedimiento almacenado	
Tarea	RF01- Definición de tareas para la creación de un proceso.	Sp_GuardarCabecera	Almacena los datos principales del encabezado del nuevo proceso: id, título, objetivo, fechaLimite, fechaInicio, fechaFin, usuario administrador, usuario supervisor y progreso.	
		Sp_GuardarTarea	Almacena cada tarea que debe realizar el Modelador, Programador, Diseñador de formularios y DBA para crear el nuevo proceso: id, idCabecera, título, descripción, tiempo de entrega, unidad de medida, usuario responsable, entre otros. Además, ejecuta automáticamente el Sp_GuardarMensaje.	
	RF02- Supervisión de tareas para diseño del proceso	Sp_ObtenerTareasNoEntregadas	Permite que el Supervisor y Administrador consulten las tareas que aún no han sido entregadas por los responsables.	
		Sp_ObtenerTareasPorIdUsuario	Permite que el Supervisor y Administrador conozcan las tareas de un usuario determinado. Y a su vez que el responsable pueda consultar su lista de tareas.	
		Sp_ObtenerTareasPorIdCabecera	Permite que el Supervisor y Administrador consulten las tareas que tienen a su supervisión o administración, respectivamente.	
		Sp_ObtenerTareasPorId	Permite consultar los campos de una tarea específica.	
		Sp_ActualizarFechas	Permite que el Supervisor cambie la fecha límite de entrega de una tarea.	
	RF03- Definición del Flujo (diagrama).	Sp_RealizarTarea	Sp_ReasignarResponsables	Permite que el Supervisor reasigne el responsable de una tarea.
			Almacena el registro de las tareas ejecutadas, dependiendo del tipo de usuario. Para el Modelador recibe el diagrama (XML), para el Programador recibe la función o procedimiento (código), para el Diseñador de formularios recibe la interfaz (HTML) y para el DBA recibe la consulta o comandos SQL (script).	
			RF04- Diseñador de formularios.	

	RF05-Integración Interna		
	RF06-Integración Externa.		
	RF07-Vista de datos.		
	RF08-Reportes.		
	RF09-Aprobar o Rechazar	Sp_AceptarORechazarTarea	Aprueba o rechaza una tarea que ha sido enviada a revisión por parte del administrador.
<b>Diseño proceso</b>	RF10-Definición de un proceso.	Sp_GuardarProceso Sp_GuardarVersion Sp_GuardarElementos Sp_GuardarCoordenadas Sp_GuardarVersionElemento	Almacena todos los elementos correspondientes al proceso, en las respectivas tablas para permitir la posterior ejecución o instancia del proceso.
	RF11-Asignación de recursos al diagrama.	Sp_GuardarAtributos	Registra los recursos (Formulario, vistas de datos, reportes, usuarios y tiempo) a la versión de un proceso.
	RF16-Rediseño de un proceso.	Sp_GuardarVersion Sp_GuardarElementos Sp_GuardarCoordenadas Sp_GuardarVersionElemento	Maneja la misma lógica del RF10-Definición de un proceso, pero en este caso no crea un nuevo proceso sino una nueva versión de un proceso existente.
		Sp_ObtenerTrámitesPorIdUsuario	Permite que el Supervisor y Administrador conozcan los trámites de un usuario determinado. Y a su vez que el Operador pueda consultar su lista de trámites.
<b>Instancia Proceso</b>	RF17-Supervisión de tareas (trámite) del proceso.	Sp_ObtenerTrámitesPorIdCabecera	Permite que el Supervisor y Administrador consulten las tareas que tienen a su supervisión o administración, respectivamente.
		Sp_ObtenerTrámitesPorId	Permite consultar los campos de un trámite específico.
		Sp_ActualizarFechaLimiteTramite	Permite que el Supervisor cambie la fecha límite de entrega de un trámite.
		Sp_ReasignarUsuarios	Permite que se pueda cambiar el responsable de un trámite.
		Sp_EliminarInstancia	Elimina todos los registros de la instancia especificada.
		Sp_ReiniciarInstancia.	Elimina todos los registros de la instancia y la deja en el punto inicial.
	Sp_ReiniciarTramite	Elimina los registros del trámite y lo vuelve al punto inicial.	
RF012-Instancia de un Proceso.	Sp_GuardarInstancia	Guarda el registro cada vez que una instancia se ejecuta por primera vez, almacenando el usuario, fecha de inicio, fecha de fin, entre otros.	
	Sp_ActualizarInstancia	Se ejecuta cada vez que se produce un cambio en la instancia. Los cambios pueden ser de estado de la instancia o del progreso, cuando un trámite perteneciente a la instancia es finalizado.	

	RF15- Historial de instancia de un proceso.	Sp_GuardarHistorialInstancia	Almacena el registro de todos los cambios de estado producidos durante la ejecución de un proceso.
		Sp_ObtenerHistorialInstancia	Devuelve el historial por instancia de un proceso.
	RF18- Ejecución de las actividades.	Sp_RealizarTrámiteManual	Almacena el registro de un trámite ejecutado de forma manual por un Operador. Ejemplo: ordenar papeles.
		Sp_RealizarTrámiteUsuario	Almacena el registro de un trámite ejecutado desde la interfaz del sistema por un Operador. Ejemplo: llenar un formulario.
		Sp_RealizarTrámiteAutomático	Ejecuta el código necesario para que una tarea automática se complete. Ejemplo: envío de un correo.
	<b>Usuario</b>	RF20- Gestión de cargos administrativos	Sp_GuardarCargos Sp_ObtenerCargos Sp_ModificarCargos

Fuente: Autores

**Cuadro 4. 18.** Procedimientos almacenados del sistema para notificaciones y reportes.

Tipo	Requerimiento Funcional	Procedimientos almacenados que intervienen	Descripción del procedimiento almacenado
<b>Notificaciones</b>	RF25- Notificaciones	Sp_GuardarMensaje	Almacena los mensajes con la descripción y ruta de las tareas que serán enviadas al Modelador, Diseñador de formularios y DBA.
		Sp_GuardarMensajeAdministrador	Almacena los mensajes con la descripción y ruta de las tareas que serán enviadas al Administrador.
		Sp_GuardarMensajeSupervisor	Almacena los mensajes con la descripción y ruta de las tareas que serán enviadas al supervisor.
		Sp_GuardarMensajeTrámites	Almacena los mensajes con la descripción y ruta de las tareas que serán enviadas a los Operadores y supervisores de proceso.
<b>Reportes</b>	RF26- Reportes	Sp_ObtenerHistorialPorInstancia	Consulta el historial registrado por instancia.
		Sp_ReporteTrámitesPorUsuario	Consulta el rendimiento de cada Operador, según los trámites finalizados o no completados a tiempo.
		Sp_ReporteInstanciasPorFechas	Consulta las instancias ejecutadas y su respectivo estado final, según un rango de fechas.
		Sp_ReporteTareasPorUsuario	Consulta el rendimiento del DBA, Diseñador de formularios o Modelador, según los trámites finalizados o no completados a tiempo.

Fuente: Autores.

**Cuadro 4. 19.** Procedimientos almacenados del sistema como herramienta de control.

Tipo	Requerimiento Funcional	Procedimientos almacenados que intervienen	Descripción del procedimiento almacenado
Control	RF16- Mecanismos de Control	Sp_SupervisiónDeTareas	Verifica la fecha límite de cada tarea, y envía un mensaje al responsable si falta 1 día para la ejecución de la misma, en caso de que se exceda del tiempo límite se notifica al Administrador y al usuario, el cual ya no podrá realizar la tarea. Esto se ejecutará automáticamente cada día a las 8 AM.
		Sp_SupervisiónDeTrámites	Verifica la fecha límite de cada trámite, y envía un mensaje al responsable si falta 1 día para la ejecución de la misma, en caso de que se exceda del tiempo límite se notifica al supervisor y al usuario, el cual ya no podrá realizar la tarea. Esto se ejecutará automáticamente cada día a las 8 AM.

Fuente: Autores.

#### 4.1.2.3. DIAGRAMA DE CLASES

- DIAGRAMA 1:** La Figura 4. 10 presenta las relaciones entre las clases que permiten crear, modificar (control de versionamiento) y eliminar (borrado lógico) los flujos de trabajo de los procesos administrativos, además de la ejecución de los mismos.

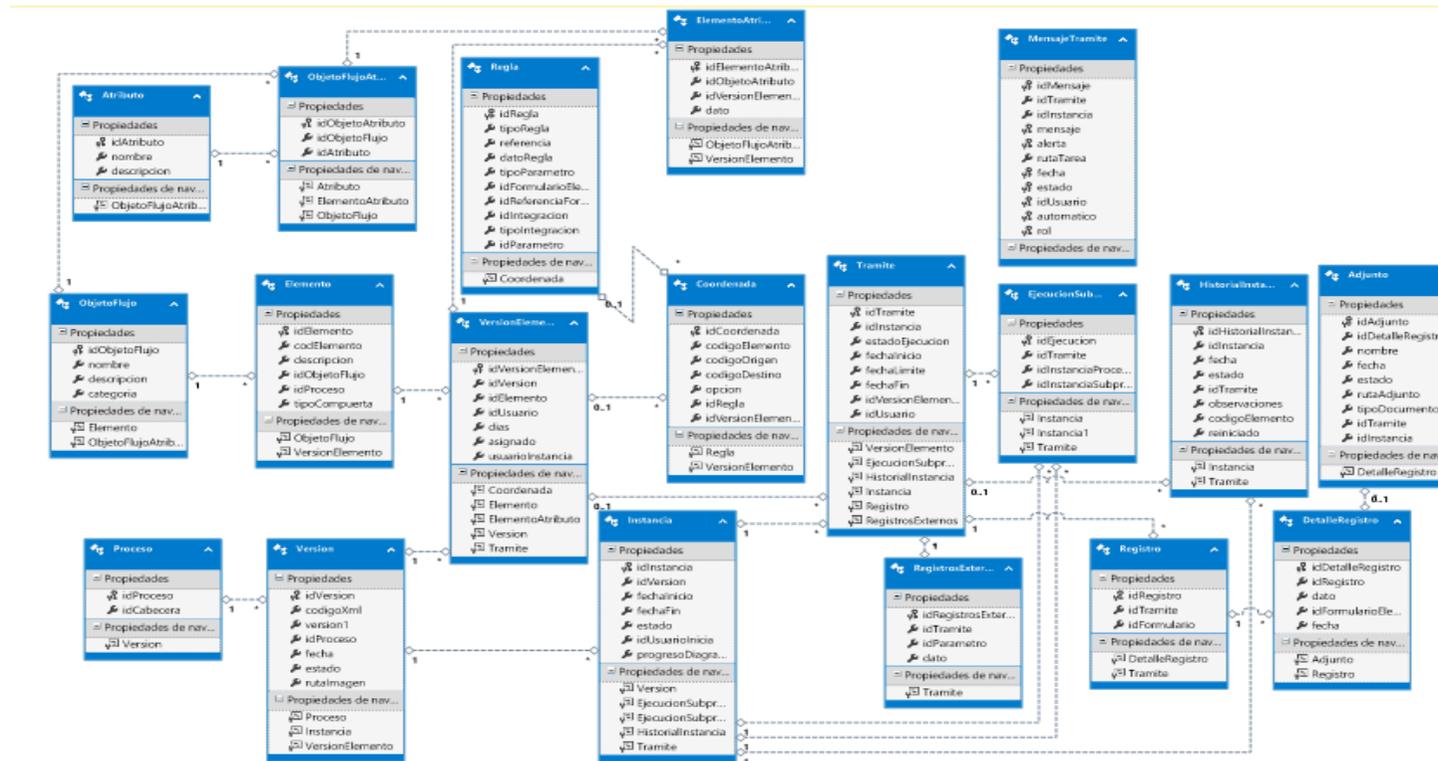


Figura 4. 10. Diagrama de clases sobre la creación y ejecución de los procesos administrativos de la ESPAM MFL.

Fuente: Autores.

#### 4.1.2.4. ENTIDADES

En el Cuadro 4. 20 se detallan las entidades que intervienen por cada requerimiento funcional. Los RF que no tengan entidad es porque fue gestionado directamente desde la BD, mediante los respectivos SP (definidos en el Cuadro 4. 18 y el Cuadro 4. 19).

**Cuadro 4. 20.** Entidades que intervienen en cada requerimiento funcional.

<b>Cód.</b>	<b>Requerimiento Funcional</b>	<b>Entidades</b>
<b>RF01</b>	Definición de tareas para la creación de un proceso.	CabeceraTarea CampoTarea Detalle Tarea Tarea
<b>RF02</b>	Supervisión de tareas para diseño del proceso.	ControlCabecera CaberaTarea
<b>RF03</b>	Definición del flujo (diagrama)	DetalleTarea CampoTarea Tarea Coordenada
<b>RF04</b>	Diseñador de Formularios.	DetalleTarea CampoTarea Tarea Formulario Formulario Elemento
<b>RF05</b>	Integración Interna (II)	DetalleTarea CampoTarea Tarea Integración Interna ParámetroIntegraciónInterna ReferencialIntegración
<b>RF06</b>	Integración Externa (IE)	DetalleTarea CampoTarea Tarea IntegraciónExterna ParámetroIntegraciónExterna
<b>RF07</b>	Vista de datos	DetalleTarea CampoTarea Tarea ReferenciaFormulario ReferenciaFormularioElemento
<b>RF08</b>	Reporte de formularios	DetalleTarea CampoTarea Tarea ReferenciaFormulario ReferenciaFormularioElemento
<b>RF09</b>	Aprobar o rechazar tareas durante el diseño del proceso.	ControlCabecera CabeceraTarea Tarea DetalleTarea CampoTarea
<b>RF10</b>	Definición de un proceso.	Proceso CabeceraTarea

		HistorialTarea
RF11	Asignación de recursos al diagrama.	Atributo Elemento Elemento Atributo Objeto Flujo ObjetoFlujoAtributo
RF12	Instancia de un proceso.	Instancia Ejecución Subproceso
RF13	Eliminar instancia de un proceso.	Instancia HistorialInstancia
RF14	Reiniciar instancia de un proceso.	Instancia HistorialInstancia
RF15	Historial de instancia de un proceso.	HistorialInstancia
RF16	Rediseño de un proceso.	Elemento Versión VersionElemento Coordenada
RF17	Supervisión de tareas (trámite) del proceso.	HistorialInstancia Trámite Instancia
RF18	Ejecución de las actividades.	Trámite Registro DetalleRegistro Registro Externo
RF19	Reiniciar actividad.	Trámite
RF20	Gestión de cargos administrativos.	Cargo Usuario
RF21	Definición de participantes.	Usuario Característica Usuario Característica
RF22	Reasignación de participantes.	Usuario Trámite Historial Instancia
RF23	Mecanismos de control	Regla Historial Instancia Historial Tarea
RF24	Errores durante la ejecución de un proceso.	Regla HistorialInstancia
RF25	Notificaciones	MensajeTarea MensajeTrámite
RF26	Reportes	Reporte
RF27	Integración de datos	IntegraciónExterna ParámetrosIntegraciónExterna ControllIntegración

Fuente: Autores.

#### 4.1.2.5.CONTROLADORES

En el Cuadro 4. 21 se detallan los controladores que intervienen por cada requerimiento funcional. Los RF que no tengan controlador es porque fue

gestionado directamente desde la BD, mediante los respectivos SP (definidos en el Cuadro 4. 18 y el Cuadro 4. 19).

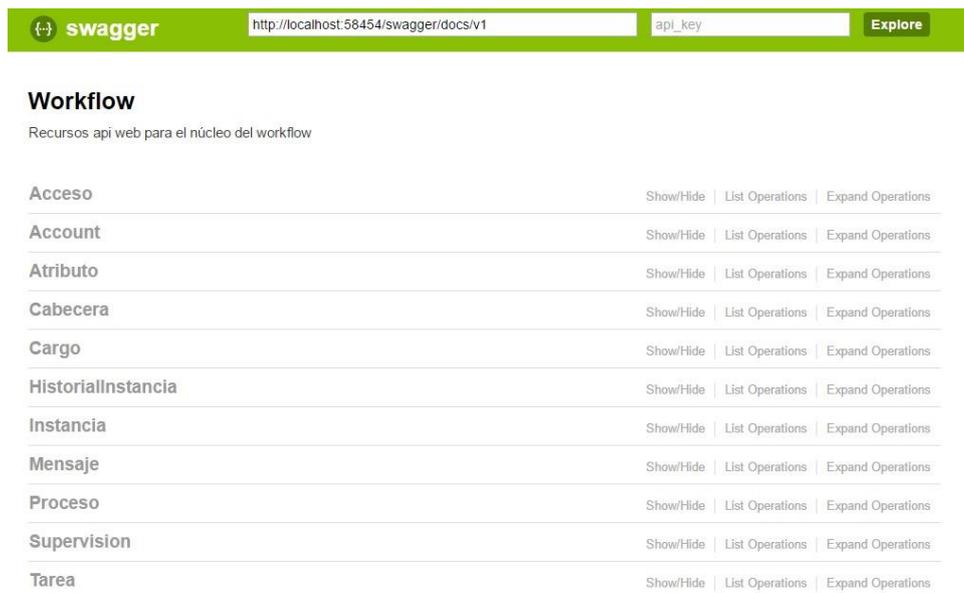
**Cuadro 4. 21.** Controladores que intervienen en cada requerimiento funcional.

<b>Cód.</b>	<b>Requerimiento Funcional</b>	<b>Controlador</b>		
RF01	Definición de tareas para la creación de un proceso.	CabeceraTarea	Tarea	
RF02	Supervisión de tareas para diseño del proceso.	Mensaje	HistorialTarea	Tarea
RF03	Definición del flujo (diagrama)	Tarea		
RF04	Diseñador de formularios.	Tarea		
RF05	Integración Interna.	Tarea		
RF06	Integración Externa	Tarea		
RF07	Vista de Datos	Tarea		
RF08	Reporte	Tarea		
RF09	Aprobar o rechazar tarea.	AprobarTarea		
RF10	Definición de un proceso.	AsignarRecurso		
RF11	Asignación de recursos al diagrama.	Regla	VersionElemento	Tarea
RF12	Instancia de un proceso.	Instancia		
RF13	Eliminar Instancia.	Instancia		
RF14	Reiniciar Instancia.	Instancia		
RF15	Historial de instancia de un proceso.	HistorialInstancia		
RF16	Rediseño de un proceso.	CabeceraTarea	Tarea	Proceso
RF17	Supervisión de tareas (trámite) del proceso.	MensajeTramite		Trámite
RF18	Ejecución de las actividades.	Registro	DetalleRegistro	
RF19	Reiniciar actividades.	Tramite		
RF20	Gestión de cargos administrativos.	Cargos		
RF21	Definición de participantes.	-		
RF22	Reasignación de participantes.	Instancia		
RF23	Mecanismos de control	-		
RF24	Errores durante la ejecución.	-		
RF25	Notificaciones	Mensaje	MensajeTramite	
RF26	Reportes	Acceso	Proceso	Usuario HistorialInstancia
RF27	Integración de datos	WebService1°		

Fuente: Autores.

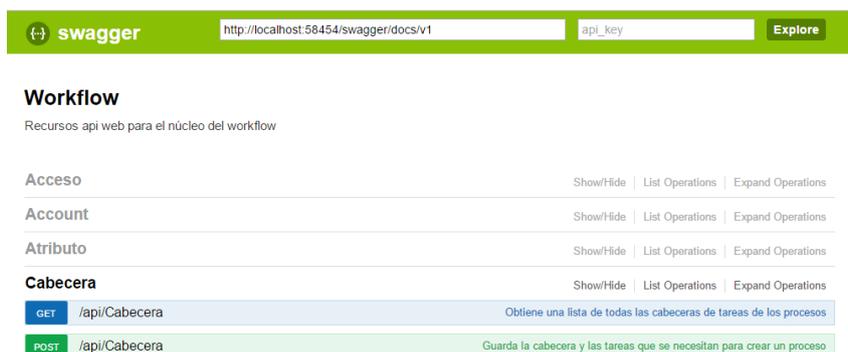
#### 4.1.2.6. INTERFAZ (RECURSOS API)

En la se muestra la interfaz que tiene la API, enlistando las entidades de los recursos disponibles, con sus posibles operaciones.



**Figura 3. 12.** Interfaz de la API, enlistando las entidades de los recursos disponibles.  
Fuente: Autores

Al seleccionar alguna entidad específica, se despliega el detalle de todos los métodos HTTP disponibles para dicha entidad. Continuando con el mismo ejemplo mencionado en la metodología (durante la etapa de desarrollo), al seleccionar “Cabecera”, en la Figura 4. 11 se pueden ver los métodos: GET y POST que pueden ser ejecutados.



**Figura 4. 11.** Métodos HTTP disponibles para CabeceraTarea.  
Fuente: Autores

De igual forma, al seleccionar un método en específico, se despliegan las posibles acciones para que se pueda ejecutar ese método. En la Figura 4. 12 se muestran detalles específicos del método POST que permite guardar una nueva cabecera con la lista de tareas. Además, un cuadro para enviar el objeto

con formato JSON y un ejemplo de cómo deben establecerse los atributos de ese elemento.

The screenshot shows a REST client interface for the 'Cabecera' resource. It displays two methods: GET and POST. The POST method is selected, and its details are shown below. The 'Parameters' section contains a table with the following data:

Parameter	Value	Description	Parameter Type	Data Type
_objCabecera	<pre>{   "_lista_tarea": [     {       "_objCabeceraTarea": {},       "_objUsuario": {         "_id_usuario": 0,         "_nombre_completo": "string",         "_cedula": "string",         "_id_identity": "string",         "_objCargo": {           "_id_cargo": 0,           "_nombre": "string"         }       },       "_email": "string"     }   ],   "_id_tarea": 0,   "_id_cabecera_tarea": 0 }</pre>	body	Model	Example Value

The 'Response Messages' section shows a table with the following data:

HTTP Status Code	Reason	Response Model	Headers
204	No Content		

The 'Parameter content type' is set to 'application/json'.

**Figura 4. 12.** Detalles para ejecutar el método POST de la Cabecera.  
Fuente: Autores

Finalmente, la Figura 4. 13 muestra la respuesta del sistema, luego de que fue ejecutado el método POST, dicha respuesta va acompañada del correspondiente código, en este caso el código 204 indica que el método se hizo de forma exitosa.

The screenshot shows the response details for the POST request. It includes three sections:

- Request URL:** `http://localhost:58454/api/Cabecera`
- Response Body:** `no content`
- Response Code:** `204`

**Figura 4. 13.** Código de respuesta para la URI api/cabecera.  
Fuente: Autores

### 4.1.3. EJECUTAR PRUEBAS DE FUNCIONAMIENTO DEL NÚCLEO.

Como se ha mencionado en el capítulo 3, las pruebas de funcionamiento se las realizó en primer lugar de manera local y se detallan en el Anexo 6. En la Figura 4. 14 se presentan dos ejemplos de las pruebas unitarias aplicadas, las pruebas de regresión no se desarrollaron ya que no existieron errores en la ejecución, además en la Figura 4. 15 se muestra la prueba funcional aplicada para la etapa de ejecución del proceso.

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_AR_01	Probar que sucede si la tarea se aprueba.	Retorna mensaje de aprobación.	Mensaje que indica al usuario que la tarea ha sido aprobada	Correcto
PU_AR_02	Probar que sucede si la tarea se rechaza.	Retorna mensaje de rechazo.	Mensaje al usuario que indica el motivo por el cual la tarea fue rechazada.	Correcto
CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_RP_01	Probar que sucede si el proceso no está listo para rediseño.	Valor booleano false, que indica la existencia de instancias en ejecución.	Retorna false.	Correcto
PU_RP_02	Probar que sucede si el proceso está listo para rediseño.	Valor booleano true, que indica que no hay instancias en ejecución.	Retorna true.	Correcto

Figura 4. 14. Ejemplos de pruebas de unidad aplicadas  
Fuente: Autores

ETAPA	CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
EJECUCIÓN	PF_EP_01	Ver Procesos disponibles.	Obtiene lista de procesos a los que puede instanciar.	Listado de proceso.	Correcto
	PF_EP_02	Iniciar instancia de proceso.	Inicia la ejecución de un proceso.	Notificación de inicio de la instancia.	Correcto
	PF_EP_03	Revisar notificación.	Listado de notificaciones.	Lista de notificaciones.	Correcto
	PF_EP_04	Ver actividad.	Obtiene datos relacionados con la actividad.	Datos de actividad.	Correcto
	PF_EP_05	Ejecutar actividad simple o de usuario.	Se guarda la actividad como realizada y se continúa con el flujo.	Notificación.	Correcto
	PF_EP_06	Actividad automática.	Notifica a los involucrados y se continúa con el flujo.	Notificación	Correcto
	PF_EP_07	Ejecutar actividad manual.	Se guarda la actividad como realizada y se continúa con el flujo.	Notificación	Correcto
	PF_EP_08	Fin de instancia.	Notificación al supervisor.	Notificación de fin de Instancia.	Correcto

Figura 4. 15. Ejemplo de prueba funcional aplicada.

Fuente: Autores

A continuación, se presenta el Cuadro 4. 22 con un resumen de la ejecución de cada recurso api indicando los códigos de respuesta y el tiempo de ejecución de cada uno, estos tiempos fueron obtenidos a través de Visual Insights.

**Cuadro 4. 22.** Tiempos de ejecución de los recursos web.

Requerimiento Funcional	Recurso Api (URI)	Método HTTP	Código de respuesta	Tiempo de ejecución
RF01. Definición de tareas para la creación de un proceso	/api/cabecera	POST	204	1.526s
RF02. Supervisión de tareas para diseño del proceso	/api/supervision	POST	204	73ms
	/api/supervision	PUT	204	80ms
	/api/tarea	DELETE	204	45ms
RF03. Definición del Flujo (diagrama)	/api/tarea	GET	204	31mss
	/api/detalletarea	POST	204	96ms
	/api/detalletarea	PUT	204	80ms
	/api/elemento	POST	204	61ms
RF04. Diseñador de Formularios	/api/filtrointegracion	GET	204	64ms
RF05. Integración interna.	/api/webservice	POST	204	30ms
RF06. Integración Externa.	/api/webservice	PUT	204	113ms
RF07. Vista de Datos.	/api/integracioninterna	GET	204	46ms
RF08. Reporte Formulario.	/api/integracioninterna	POST	200	64ms
	/api/integracioninterna	PUT	204	17ms
RF09. Aprobar o rechazar tareas durante el diseño del proceso.	/api/integracionexterna	GET	204	40ms
	/api/integracionexterna	POST	204	30ms
RF10. Definición de un proceso.	/api/referenciaformulario	GET	204	32ms
	/api/referenciaformulario	POST	200	36ms
RF11. Asignación de recursos al diagrama.	/api/filtrotarea	GET	204	34ms
	/api/aprobartarea	POST	204	121ms
RF12. Instancia de un proceso.	/api/asignarrecurso	GET	204	488ms
RF13. Eliminar Instancia de un proceso.	/api/versioelemento	PUT	204	57ms
	/api/regla	GET	204	97ms
	/api/regla	POST	200	41ms
RF14. Reiniciar Instancia de un proceso.	/api/instancia	POST	204	1 267s
RF15. Historial de instancia de un proceso.	/api/instancia	DELETE	204	1 345s
	/api/instancia	PUT	204	2 120s
	/api/instancia	GET	204	312ms
RF16. Rediseño de un proceso.	/api/historialinstancia	GET	204	13ms
	/api/historialinstancia	POST	200	67ms
	/api/filtroredisenho	GET	204	124ms
RF17. Supervisión de tareas (trámite) del proceso.	/api/proceso	POST	204	230ms
	/api/proceso	PUT	204	125ms
	/api/responsable	GET	204	45ms
RF18. Ejecución de las actividades.	/api/instancia	PUT	204	23ms
	/api/tramite	PUT	204	54ms
	/api/ejecucionatm	GET	204	452ms
	/api/ejecucionformulario	POST	204	568ms
RF18. Ejecución de las actividades.	/api/ejecucionvdfcompuerta	POST	204	450ms
	/api/ejecucionvdf	POST	204	235ms
	/api/ejecucionvdi	POST	204	462ms
	/api/ejecucionvdi	POST	204	452ms

RF19. Reiniciar actividad.	/api/tramite	PUT	204	235ms
RF20. Gestión de cargos administrativos.	/api/cargo	GET	204	230ms
	/api/cargo	POST	200	12ms
	/api/cargo	PUT	200	20ms
	/api/cargo	DELETE	200	14ms
RF21. Definición de participantes.	Se ejecuta de manera interna, es decir que no interviene ningún controlador.			
RF22. Reasignación de participantes.	/api/responsable	GET	204	45ms
RF23. Mecanismos de control.	Los mecanismos de control implementados, corresponden a la supervisión de las tareas del proceso, en las etapas de diseño y ejecución. Además se hace uso del Agente de SQL para la verificación de las fechas límite de las tareas.			
RF24. Errores durante la ejecución de un proceso.	Los errores que se presentan en la ejecución de un proceso, están especificados en la definición de este requerimiento en el anexo 2, además la solución de los mismos se presentan en los requerimientos 13, 14 y 19.			
RF25. Notificaciones.	/api/mensaje	GET	204	11ms
	/api/mensajetramite	GET	204	15ms
	/api/filtromensaje	GET	204	36ms
	/api/filtromensajetramite	GET	204	36ms
	/api/filtromensajetramite	POST	204	36ms
	/api/filtroultimasalertas	GET	204	25ms
RF26. Reportes.	/api/reporte	GET	204	20ms
	/api/reporte	POST	200	19ms
	/api/documentacion9001	GET	204	121ms
RF27. Integraciones	Las integraciones pueden ser internas (BD de la aplicación) o a Base de datos externas, los recursos utilizados se implementan en los requerimientos 5, 6, 7 y 8.			

**Elaboración:** Autores.

#### 4.1.4. IMPLEMENTAR EL NÚCLEO EN LOS SERVIDORES DE LA INSTITUCIÓN

##### 4.1.4.1. CARACTERÍSTICAS GENERALES DEL SISTEMA

Luego de llevar a cabo todas las etapas de MEIDAW, detalladas en el capítulo anterior, se pudo implementar un sistema de calidad que facilita la automatización de los procesos administrativos de la ESPAM MFL y que contiene las siguientes características:

- Permite el diseño, ejecución y seguimiento de los procesos administrativos de la institución. Además, presenta flexibilidad para que se adapten los procesos a los cambios que se puedan presentar a través del tiempo, mediante el rediseño.

- Administra la ejecución automática de procesos, mediante la interpretación del diseño o definición previa de los mismos.
- Contiene un mecanismo para crear múltiples instancias de los procesos y de forma análoga, permite la ejecución de actividades que se alcanza mediante la instancia de los pasos lógicos (trámites) del flujo del proceso.
- Durante la ejecución, interactúa con los recursos previamente establecidos en la etapa de diseño. Considerando que un recurso es el ente que actúa en la organización para llevar a cabo los ítems de trabajo asignados y que cada ítem es la representación de una tarea que ha sido determinada para completar un paso lógico del flujo de trabajo, lo cual puede interpretarse como la ejecución de una actividad perteneciente a la instancia de un proceso.
- Permite la generación de reportes con flexibilidad de combinaciones de campos/parámetros de información, según las necesidades de la entidad.
- Presenta seguridad de acceso para garantizar la confidencialidad e integridad de la información.
- Permite que se realice el seguimiento adecuado por parte de los interesados, durante las etapas de diseño y ejecución de los procesos.

#### **4.1.4.2.ESQUEMA DE FUNCIONAMIENTO DEL SISTEMA**

El desarrollo del núcleo se llevó a cabo de acuerdo a las recomendaciones de la WFMC en su documento dedicado al Modelo de Referencia para los sistemas Workflow escrito por Hollingsworth (1995) y citado por Menéndez y Castellanos (2016). Tal como su autor lo expresa, no fue necesario seguir todas las indicaciones para definir la perspectiva organizacional del sistema, pero fue una guía sólida para establecer la forma en que opera el motor del Workflow. A continuación, en la Figura 4. 16 se presenta un esquema del “modo operandi” o forma de trabajar del núcleo desarrollado por los autores de esa tesis, en cuanto al diseño y ejecución de procesos. Las clases en el diagrama, a excepción de Ítem de Trabajo, son los paquetes bases para el funcionamiento del sistema, mismos que son explicados posteriormente de forma detallada.

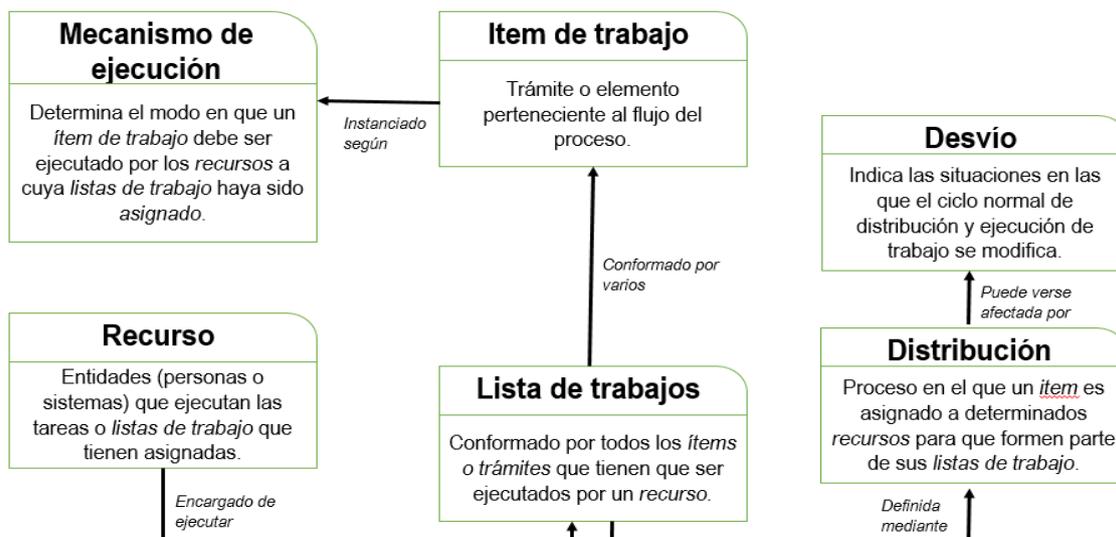


Figura 4. 16. Esquema de ejecución del núcleo del workflow.

Fuente: Autores

- **El Paquete Recursos:** son los encargados de realizar los ítems o trámites pertenecientes a un flujo de trabajo. Dentro del sistema se puede definir como recurso a un participante humano (usuario Operador) para que lleve a cabo Tareas Simples, de Usuario o Manuales o se puede asignar entidades de software, para que realicen las Tareas Automáticas (generación de reportes).
- **El Paquete Distribución:** siendo el mecanismo por el cual un trámite o Ítem de trabajo es asociado con un recurso para su ejecución, se pueden definir dos tipos de distribución:
  - La asignación por usuario específico o entidad de software: en la que el ítem es asociado con la lista de trabajo de un recurso en concreto. Esta distribución se realiza en tiempo de diseño del proceso.
  - La oferta: donde se asocia de forma voluntaria el Ítem a la lista de trabajo, la cual se realiza mediante la asignación por cargo administrativo (estructura organizacional); se permite que cualquier usuario que tiene el cargo fijado para realizar el trámite, tome la libre decisión de realizarlo. La distribución del cargo se realiza en etapa de diseño, pero la definición del usuario voluntario se realiza en tiempo de ejecución.

- **El Paquete Mecanismos de Ejecución:** luego de que se han distribuido todos los ítems a sus correspondientes recursos, se pueden ejecutar mediante los siguientes tipos específicos de mecanismos de ejecución:
  - En cadena: cuando el trámite actual ha finalizado, automáticamente se ejecuta el trámite siguiente.
  - Iniciada por recurso: en las compuertas, cuando el trámite actual tiene responsable un usuario que debe decidir qué camino seguir en el flujo, se ejecutará el trámite seleccionado, en cambio, si tiene asignada una regla, se ejecutará el siguiente trámite que cumpla la condición de la misma.
  - Apilada: cuando ha terminado una instancia con un evento de fin de mensaje que contiene un código único de proceso que indicará el proceso que debe ejecutarse de forma automática, mismo que debe tener un evento de inicio de mensaje.
- **El Paquete Desvíos:** luego de que el ítem ha sido asignado, o luego de que ha iniciado su ejecución, pueden presentarse situaciones que requieran modificar la distribución realizada o volver a realizar la ejecución. Los tipos de desvíos son:
  - Delegación: cuando el trámite no está siendo ejecutado, pero se cambia de responsable.
  - Re-ejecución: cuando un trámite no fue finalizado a tiempo y es necesario que sea reiniciado, o en su defecto, se ha presentado un error que requiere que la instancia de todo el proceso sea reiniciado.
  - Re-asignación: el trámite está siendo ejecutado, pero se asigna otro responsable, el cual debe volver a hacer todo desde el inicio, ya que es una reasignación sin estado.
  - Finalización por error: permite terminar de forma automática la ejecución, porque se ha presentado un error que no puede ser controlado durante la instancia.
- **El Paquete Lista de Trabajos:** contiene todos los trámites o ítems de trabajo que deben ejecutar los recursos durante la instancia de los procesos. Las listas pueden ejecutarse de dos formas:
  - El usuario puede decidir el orden en el que va a ejecutar las tareas que tiene asignadas.

- El sistema ejecuta las tareas asignadas a los recursos automáticos, de acuerdo a lo establecido en el flujo de trabajo.

## 4.2. DISCUSIÓN

Dado que la ESPAM MFL es una institución que maneja una gran cantidad de procesos administrativos, fue necesario desarrollar el núcleo del Workflow para facilitar la automatización de dichos procesos. Para lo cual, como punto inicial fue imprescindible determinar la metodología de desarrollo, tal como lo afirman Cervantes y Gómez (2012) en su artículo sobre la taxonomía de los modelos y metodologías de desarrollo de software más utilizados, destacando que el éxito de los proyectos de software depende en gran medida de que tengan un buen inicio, siendo uno de los factores claves la elección de la metodología.

Partiendo de los tipos de metodologías: ágiles y tradicionales, Jiménez y Orantes (2012) también especifican que se han creado las metodologías híbridas, las cuales consideran algunas de las mejores características de ambas metodologías. Reflexionando en este sentido, dichos autores en su investigación sobre la necesidad de utilizar una metodología híbrida para el desarrollo de software en México, establecen que es necesario que la misma pueda proveer un conjunto de prácticas y herramientas que faciliten el proceso de desarrollo, ofreciendo un producto con alta calidad, seguro y que satisfaga las expectativas del cliente. En este sentido, Mendoza y Reynoso (2001) en su artículo titulado MEIDAW: una propuesta metodológica para mejorar el proceso de desarrollo de sistemas Workflow, destacan que dicha metodología ha sido diseñada para que se adapta a las necesidades de la programación de dichos sistemas, misma que fusiona elementos de MEI en el enfoque al proceso, y el soporte de los estándares metodológicos de la WfMC, para obtener la calidad del producto.

Por otra parte, Marrero (2011) en su tesis doctoral propone una nueva metodología ágil para el desarrollo de Workflows, la cual es una modificación de Scrum y se basa principalmente en esclarecer algunos de sus puntos, sobre

todo lo relacionado a la gestión de las reuniones de trabajo. La principal modificación es la implementación de una nueva fase denominada diseño de workflows, donde se realiza el diagrama del proceso que se va a automatizar, sin considerar lo recomendado por la WfMC, a diferencia de MEIDAW, la metodología utilizada en la ejecución de este trabajo, que toma en cuenta estas especificaciones durante la segunda fase: desarrollo del producto.

Y desde el aspecto de desarrollo, se analiza lo realizado por Alulema (2011) durante el análisis, diseño e implementación de una aplicación Workflow para el seguimiento de procesos de los servicios que ofrece el área de comercialización de la Empresa Eléctrica Regional Centro Sur, el cual consta de dos procesos específicos: recepción de solicitud de servicio e inspección de la misma. Este sistema permite a los clientes de la institución la solicitud de un nuevo servicio de energía eléctrica, con su respectiva inspección, entre otros aspectos. Sin embargo, no soporta la definición de nuevos roles de usuario, creación de nuevos procesos o aplicación de reingeniería a los existentes, mientras que el núcleo desarrollado por los autores de esta tesis, admite la ejecución de dichas acciones, mediante: definición de nuevos cargos a todos los usuarios con el Rol Operador; creación de nuevos procesos administrativos, a través de: diseño de diagramas de procesos (según el estándar Business Process Manager Notation BPMN 2.0), definición de usuarios responsables para las actividades, creación de nuevas validaciones e integraciones de datos, además del respectivo control de la ejecución de los procesos; actualización de los procesos existentes, a través de los respectivos recursos de versionamiento.

# **CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES**

## **5.1. CONCLUSIONES**

Luego de alcanzar el desarrollo del núcleo del Workflow, los autores plantearon las siguientes conclusiones:

- La elaboración de un plan de proyecto en la fase inicial de este trabajo, permitió obtener principalmente: los requerimientos generales del núcleo, el esquema general de funcionamiento y las herramientas tecnológicas a utilizar. Con dichos requerimientos y mediante el uso del estándar IEEE830 se pudo identificar el alcance, así como los requerimientos funcionales del core.
- El software desarrollado permite que los integrantes de la UPS, encargados de sistematizar los procesos académicos de la ESPAM MFL, puedan automatizar los procesos administrativos de dicha institución, accediendo por medio de autenticación basada en TOKENS, a los recursos web brindados por el sistema, de forma sencilla y segura.
- Se desarrolló un núcleo de calidad, que cumple lo establecido por la WfMC, dado que fue desarrollado bajo las directrices dadas por dicha organización y evaluado a través de un plan de pruebas, documento fundamental que permitió detectar todas las posibles falencias en las respuestas brindadas por el software, y a su vez, la correspondiente gestión de incidencias, facilito la corrección de los mismos.
- El software implementado en los servidores de la ESPAM MFL fue desarrollado para que la institución pueda consumir los recursos que permiten diseñar, ejecutar y supervisar sus procesos administrativos, permitiendo que lleven un control de calidad de las actividades, ya sean manuales o automáticas, que componen los procedimientos automatizados mediante el núcleo de la herramienta Workflow.

## 5.2. RECOMENDACIONES

De acuerdo a las conclusiones anteriores, se recomienda lo siguiente:

- Una vez realizado el levantamiento de información, es importante conocer las técnicas propuestas por la metodología empleada, para identificar el alcance y requerimientos funcionales del proyecto. Se recomienda la utilización del estándar IEEE830 como una de las mejores prácticas para dicha etapa. Además, es elemental conocer las ventajas y características que ofrecen las herramientas a utilizar, y verificar que estas permitan alcanzar el desarrollo de los requerimientos estipulados.
- Previo a la codificación, es necesario realizar un análisis de cómo se pretende que funcionará el software, para obtener un esquema de base de datos óptimo, ya que de ello dependerá el correcto funcionamiento del mismo y a su vez contribuirá a la creación de una API WEB de calidad, agilizando los tiempos de respuesta a las peticiones HTTP, enviadas por parte de los clientes.
- Otro tema indispensable igual que el rendimiento, es la seguridad, por lo cual se recomienda que durante la construcción de los recursos web, se implementen mecanismos de autenticación y encriptación, además de los brindados por el SGBD, a fin de alcanzar un sistema eficaz e íntegro.
- Antes de realizar las pruebas de funcionamiento, se debe elaborar un plan de pruebas donde se especifiquen los aspectos necesarios para cada caso de prueba (entradas) y los resultados que se esperan alcanzar (para poder comparar y verificar las salidas esperadas con las obtenidas). Además es necesario llevar a cabo la correspondiente gestión de incidencias, que puede servir como guía para futuros trabajos, indicando los problemas que se presentaron y la forma en que fueron solucionados.
- Es fundamental que las organizaciones cambien el enfoque del tratamiento de las actividades y recursos utilizados durante el diseño y ejecución de sus procesos, para que puedan adaptarlo a las ventajas y oportunidades que brindan los Sistemas Gestores de Flujo de Trabajo (BPMS), a fin de que logren aprovechar la inversión requerida, consiguiendo que puedan ser modelados, ejecutados y supervisados con mayor eficacia y eficiencia.

## BIBLIOGRAFÍA

- Aguilar, M. 2012. Aprendizaje y Tecnologías de Información y Comunicación: Hacia nuevos escenarios educativos. CO. Revista Latinoamericana de Ciencias Sociales, Niñez y Juventud. Vol. 10. Núm. 2. p 801-811
- Alayón, J. 2016. API de servicios web REST extensible dinámicamente basada en una arquitectura Plug-in para el procesado de imágenes recibidas desde dispositivos móviles. Tesis. Ing. en Informática. Escuela Técnica Superior de Ingeniería Informática de la Universidad de La Laguna. Santa Cruz de Tenerife, ES. p 24-33.
- Alulema, D. 2011. Análisis, diseño e implementación de una aplicación Workflow para el seguimiento de procesos de los servicios que ofrece el área de comercialización de la Empresa Eléctrica Regional Centro Sur. Tesis. Ingeniero de Sistemas. Universidad Politécnica Salesiana. Cuenca-Azuay, EC. p 203.
- Amodeo, E. 2013. Principios de diseño de APIs REST. España. Leanpub. p 3-60.
- Arbeláez, O; Medina, F; Chaves, O. 2011. Herramientas para el desarrollo rápido de aplicaciones web. Pereira, CO. Scientia Et Technica. Vol. 17. p 254 - 258.
- Artiles, S. y Márquez, Y. 2013. El Modelo de Gestión de Información y Conocimiento: resultados de su aplicación en una empresa en perfeccionamiento. CU. Revista Internacional de Gestión del Conocimiento y la Tecnología. Vol. 1. Núm. 1 p 13-23.
- Avellán, N y Zambrano, T. 2015. Aplicación web de la gestión del programa semillero de investigadores de la ESPAM MFL. Tesis. Ingeniero en Informática. ESPAM MFL. Calceta-Manabí, EC. p 15.
- Brice-Arnaud, G. 2016. ASP.NET en C# con Visual Studio 2015: Diseño y desarrollo de aplicaciones web. En ediciones. p 384.
- Calle, X; Mayorga, F; Flores, A; Lavín, J. 2016. Aplicación de la metodología BPM: RAD en una institución de educación superior. Cuenca, EC. Revista Maskana. Vol 65. p 223-224.
- Ceballos, F. 2012. Enciclopedia de Microsoft Visual C#: interfaces gráficas y aplicaciones para Internet con Windows Forms y ASP.NET. 4a. ed. RAMA Editorial.
- CENDITEL (Centro Nacional de Desarrollo e Investigación en Tecnologías Libres). 2013. Aseguramiento de calidad en el desarrollo de software libre. VE. p 27.
- Cervantes, J. y Gómez, M. 2012. Taxonomía de los modelos y metodologías de desarrollo de software más utilizado. Unión de Universidades de América

Latina y el Caribe Distrito Federal, Organismo Internacional. Revista Universidades. Vol. 62. p 37-47.

Cruz, J. y Lévano, D. 2013. Nivel de madurez de los procesos de la gestión de servicios en base a BPM. PE. Revista Apuntes Universitarios. Núm. 1. p 143-156.

Delgado, A. y Calegari, D. 2014. Aspectos clave para la construcción de software de soporte e impacto en la mejora continua de las organizaciones. UY. Revista INNOTEC Gestión. Vol. 6. p 40-51.

De Soto, A. y Fernández, E. 2006. Nuevas Tendencias en Sistemas de Información: Procesos y Servicios. ES. Revista Pecunia. Vol. 2. p 129-158.

Derecho Ecuador. 2012. Registro Oficial N° 318: Estatuto orgánico de gestión organizacional por procesos. (En línea). EC. Consultado, 10 de nov. 2016. Formato HTML. Disponible en <http://www.derechoecuador.com/productos/producto/catalogo/registros-oficiales/2012/julio/code/20397/registro-oficial-no-318--lunes-30-de-julio-del-2012-edicion-especial>

Dietz, N. 2016. Fulfilling a RESTful Commitment: Test-Suite Generation for Swagger-based REST APIs. California, US. p 1-38.

ESPAM-MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López). s.f. Reseña Histórica de la ESPAM MFL. (En línea). EC. Consultado, 02 de jun. 2016. Formato PDF. Disponible en <http://espam.edu.ec>

Feliciano, S; Serrano, J; Feliciano, A. 2012. Desarrollo de una herramienta matemática como apoyo a la educación básica en el estado de Guerrero (México) utilizando Monodevelop con C#. Revista Vínculos Vol 9. p 95-113.

Flores, A; Lavín, J; Calle, X; Álvarez, E. 2014. Buscando la excelencia educativa: Gestión de procesos académicos y administrativos en Instituciones Públicas de Educación mediante BPM. Cuenca, EC. Revista Maskana. Vol. 65. p 199-209.

Flores, D. y Rufino, J. 2007. Implementación de un Sistema de Gestión del Flujo de Trabajo de los Procesos Administrativos del área de Promoción y Afiliación de la Sociedad Nacional de Industrias – SNI sede Lambayeque. Tesis. Ing. de Sistemas. Universidad Señor de Sipán. Chiclayo, PE. p 60-81.

Fotache, M. & Strimbei, C. 2015. SQL and Data Analysis. Some Implications for Data Analysis and Higher Education. RO. Journal Procedia Economics and Finance. Vol. 20. p 243-251.

- Fresno, C. 2010. Workflows flexibles para procesos de desarrollo de software. Tesis. Máster en Ingeniería del Software, Métodos Formales y Sistemas de Información. Universidad Politécnica de Valencia. Valencia, ES. p 145
- Galvão, M. 2012. Manual de software: facilitando a comunicação entre empresa produtora e sociedade. Transinformação. Vol. 19. p 45-64.
- Jiménez, E y Orantes, S. 2012. Metodologías híbridas para desarrollo de software: una opción factible para México. MX. Revista Digital Universitaria. Vol. 13. p 1-16.
- Karabogolian, L; Bazán, P; Martínez, J. 2014. Ejecución y monitoreo de procesos de negocios distribuidos entre diferentes motores de Bonita OS. In XVI Workshop de Investigadores en Ciencias de la Computación. p 420-424.
- Llanes-Font, M; Isaac-Godínez, C; Moreno-Pinol, M; García-Vidal, G. 2014. De la gestión por procesos a la gestión integrada por procesos. CU. Revista Ingeniería Industrial. Vol. 25. Núm. 3. p 255-264.
- Marciszack, M; Pérez, M; Groppo, M. 2015. Metodología y herramienta de soporte para validar modelos conceptuales a través de máquinas abstractas. Buenos Aires, AR. Revista Ciencia y Tecnología. Vol. 15. p 165-180.
- Marrero, M. 2015. Sistemas "Workflow" y BPM ("Business Process Management") como herramientas para la automatización y mejora de la productividad en las organizaciones: metodología para la implantación y casos de estudio. Tesis. Doctorado en Ciencias Económicas y Empresariales. ULPGC. Gran Canaria, ES. p 39.
- Martín, M; Leyva, G; Peña, N; Labrada, R. 2016. Proceso de réplica de datos con Microsoft SQL Server para el Replicador de Datos Reko. CU. Revista Cubana de Ciencias Informáticas. Vol. 10. p 128-142.
- Mendoza, L. y Reynoso, W. 2001. MEIDAW: Una Propuesta Metodológica para Mejorar el Proceso de Desarrollo de Sistemas Workflow. San Salvador de Jujuy, AR. Editorial Universidad Nacional de Jujuy. p 99-108.
- Menéndez, V. y Castellanos, M. 2016. Los Sistemas Gestores de Flujos de Trabajo en la Gestión de Procesos Software. Revista Computación e Informática. Vol. 5. Núm. 3.
- Microsoft. 2015. ASP.NET Web API. (En línea). Consultado, 12 de Oct. 2016. Formato HTML. Disponible en <https://msdn.microsoft.com>
- Microsoft. 2017. SQL SERVER AGENT. (En línea). Consultado, 8 de Ago. 2017. Formato HTML. Disponible en <https://docs.microsoft.com/es-es/sql/ssms/agent/sql-server-agent>

- Morales, A; Escoto, M; García, R; Molinar, J; Hidalgo, C. 2012. Sistema para la Aplicación de Pruebas Psicológicas vía Web Acta Universitaria. Universidad de Guanajuato. REDALYC Red de Revistas Científicas de América Latina, el Caribe, España y Portugal Sistema de Información Científica. Vol. 22 N° 3, pp. 5-13
- Mujumdar, A; Masiwal, G; Chawan, P. 2012. Analysis of various software process models. Mumbai, IN. International Journal of Engineering Research and Applications. Vol. 2. p 2015-2021.
- Peralta, M; Salgado, S; Berón, M; Riesgo, D; Montejano, G; Baigorria, L. 2013. Aplicación de Técnicas de Comprensión de Programas para la Obtención de Información de Modelos de Procesos Workflow. AR. Departamento de Informática. p 556-561
- Registro Oficial. 2013. Norma técnica de administración por procesos. (En línea). Consultado, 10 de nov. 2016. Formato PDF. EC. Disponible en: <http://www.diccionario.informatica.gob.ec/adjuntos/norma-tecnica-de-administracion-por-procesos.pdf>
- Rivadeneira, G. 2013. Metodologías ágiles enfocadas al modelado de requerimientos. AR. Revista Informe Científico Técnico UNPA. Vol. 5. Num. 1. p 1-29.
- Robledo, P. 2011. El Libro del BPM: introducción. España. p 5-8.
- Rodríguez, E. 2011. Implementación de BPM, como herramienta de integración y administración de una organización. Tesis. Ing. Informática. Facultad de Informática. Universidad Técnica Particular de Loja. Loja, EC.
- Salazar, Z. y Prado-Calderón, J. 2013. Importancia de la planificación sobre la recolección de los datos: aprendizajes a partir de una experiencia de investigación. CR. Revista Ciencias Sociales. Num. 141. p 35-47.
- SENPLADES (Secretaría Nacional de Planificación y Desarrollo). 2013. Plan Nacional del Buen Vivir: Objetivo 1: Consolidar el Estado democrático y la construcción del poder popular. (En línea). EC. Consultado, 1 de oct. 2016. Formato HTML. Disponible en <http://www.buenvivir.gob.ec/objetivo1#tabs2>
- Stein, M. & Geyer-Schulz, A. 2013. A Comparison of Five Programming Languages in a Graph Clustering Scenario. Karlsruhe, DE. Journal Universal Computer Science. Vol. 19. p 428-456.
- Trivedi, S. 2012. Software testing techniques. IN. International Journal of Advanced Research in computer science and software Engineering. Vol. 2. Num. 10.
- Varanasi, B. y Belida, S. 2015. Spring REST: documenting REST Services. Apress. p 91-104.

Workflow Management Coalition. 2014. ¿Qué es BPM? Consultado, 25 de Jul. 2016. (En línea). Disponible en: <http://www.wfmc.org/what-is-bpm>

Zorzan, F; Daniele, M; Frutos, M; Uva, M. 2014. Una herramienta para la automatización de procesos de desarrollo de software usando QVT: transformación de controles de flujo SPEM a BPMN. XX Congreso Argentino de Ciencias de la Computación. Buenos Aires. p 20.

# **ANEXOS**

ANEXO 1 – PLAN DE PROYECTO



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

**WfMC**

**Núcleo de un Workflow de  
Procesos Administrativos de la ESPAM MFL**



**PLAN DE  
DESARROLLO DE  
SOFTWARE**

Dayana H. Bailón Delgado  
José S. Mera Cantos

**2017**

## FICHA DEL DOCUMENTO

El cuadro 1 detalla las versiones que ha tenido este anexo, indicando: fecha, versión, autores y por quien fue revisado.

**Cuadro 1.** Ficha del documento.

<b>Fecha</b>	<b>Versión</b>	<b>Autores</b>	<b>Verificado por</b>
<b>22/11/2016</b>	1.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira
<b>23/01/2017</b>	2.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira
<b>10/08/2017</b>	3.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira

## CONTENIDO GENERAL

FICHA DEL DOCUMENTO .....	115
CONTENIDO GENERAL.....	116
CONTENIDO DE CUADROS Y FIGURAS.....	118
1. INTRODUCCIÓN.....	119
1.1. PROPÓSITO.....	119
1.2. ALCANCE .....	119
1.3. RESUMEN .....	120
2. VISTA GENERAL DEL PROYECTO .....	120
2.1. PROPÓSITO, ALCANCE Y OBJETIVOS.....	120
2.2. SUPOSICIONES Y RESTRICCIONES .....	121
2.3. ENTREGABLES DEL PROYECTO.....	121
3. ORGANIZACIÓN DEL PROYECTO .....	122
3.1. PARTICIPANTES EN EL PROYECTO .....	122
3.2. INTERFACES EXTERNAS .....	123
3.3. ROLES Y RESPONSABILIDADES .....	123
4. GESTIÓN DEL PROCESO .....	123
4.1. ESTIMACIONES DEL PROYECTO .....	123
4.2. PLAN DEL PROYECTO.....	125
4.2.1. PLAN DE FASES.....	125
4.2.2. CRONOGRAMA DEL DESARROLLO DEL PROYECTO.....	125
4.3. SEGUIMIENTO Y CONTROL DEL PROYECTO .....	126
4.3.1. GESTIÓN DE REQUISITOS.....	126
4.3.2. CONTROL DE PLAZOS .....	126
4.3.3. CONTROL DE CALIDAD.....	126
5. ESPECIFICACIONES DEL PRODUCTO .....	127
5.1. ESQUEMA DE FUNCIONAMIENTO DEL NÚCLEO.....	127
5.2. PILA PARA EL DESARROLLO.....	128
5.2.1. LENGUAJE DE PROGRAMACIÓN .....	128
5.2.2. ENTORNO DE DESARROLLO INTEGRADO (IDE).....	131
5.2.3. FRAMEWORK.....	131
5.2.4. SISTEMA GESTOR DE BASE DE DATOS (SGBD).....	132

6. GLOSARIO Y ABREVIATURAS .....	133
6.1. GLOSARIO .....	133
6.2. ABREVIATURAS.....	134
7. REFERENCIAS .....	135

## CONTENIDO DE CUADROS Y FIGURAS

<b>Cuadro 1.</b> Ficha del documento.....	115
<b>Cuadro 2.</b> Lista de entregables.....	122
<b>Cuadro 3.</b> Roles y responsabilidades. ....	123
<b>Cuadro 4.</b> Presupuesto de desarrollo. ....	123
<b>Cuadro 5.</b> Plan de fases de proyecto. ....	125
<b>Cuadro 6.</b> Cronograma de desarrollo del proyecto.....	125
<b>Cuadro 7.</b> Requerimientos generales del núcleo.....	128
<b>Cuadro 8.</b> Los cinco lenguajes de programación más populares. ....	129
<b>Cuadro 9.</b> Tabla comparativa entre Java, C# y Python. ....	129
<b>Figura 1.</b> Esquema general de la gestión de los procesos administrativos. ..	127
<b>Figura 2.</b> Esquema de funcionamiento general del núcleo.....	128

## **1. INTRODUCCIÓN**

Este Plan de Desarrollo de Software provee una visión global del enfoque de desarrollo propuesto.

El proyecto está basado en la Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow (MEIDAW), donde se indica como punto base o de partida, el desarrollo de un Plan de Proyecto que permita identificar los requerimientos del software, las herramientas de desarrollo, los objetivos y estructura del proyecto, los entregables que serán generados, entre otros aspectos. Este documento es a su vez uno de los entregables de MEIDAW.

### **1.1. PROPÓSITO**

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto — lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- Los miembros del equipo de desarrollo — lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

### **1.2. ALCANCE**

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL”. El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. La versión 2.0 del Plan de Desarrollo del Software se ha realizado con base al levantamiento de información, mediante investigación bibliográfica por parte de

los autores y a las correcciones de la versión 1.0 realizadas por el tutor. El avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento, produciendo nuevas versiones actualizadas.

### **1.3. RESUMEN**

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

- Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los entregables que serán producidos y utilizados durante el mismo.
- Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.
- Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.
- Especificaciones del Producto — proporciona una vista global del funcionamiento del producto y determina las herramientas que serán utilizadas.
- Glosario y Abreviaciones — especifica la definición de los términos y abreviaciones utilizadas en el documento.
- Referencias — detalla las fuentes bibliográficas utilizadas.

## **2. VISTA GENERAL DEL PROYECTO**

### **2.1. PROPÓSITO, ALCANCE Y OBJETIVOS**

La ESPAM MFL ejecuta diferentes procesos administrativos, a través de documentos, información o tareas, que son transmitidos de un integrante a otro, de acuerdo a un conjunto de reglas establecidas, dentro de la organización. Aunque actualmente tales procesos se gestionan de forma manual, los autores del proyecto pretenden facilitar su automatización mediante el desarrollo del núcleo de un Workflow.

El objetivo del núcleo es procesar la información de las actividades que intervienen en los diferentes procesos administrativos para asignarlas a los recursos correspondientes, ya sean humanos o tecnológicos, a fin de que cumplan con las respectivas tareas: manuales o automáticas.

## **2.2. SUPOSICIONES Y RESTRICCIONES**

- La automatización de los procesos administrativos se realizará con base a lo establecido en el Reglamento de Procesos de la ESPAM MFL.
- La información registrada por parte de los usuarios finales es verídica.
- El núcleo del Workflow deberá integrarse posteriormente con las respectivas interfaces, desarrolladas por las autoras de una tesis complementaria.
- Las restricciones técnicas como lenguaje de programación, Sistema Gestor de Base de Datos (SGBD), requisitos del servidor, entre otros, serán profundizadas en el Anexo 2: Documento de Especificación de Requerimientos de Software (IEEE 830).
- El mantenimiento del sistema será realizado por parte de los integrantes de la Unidad de Producción de Software (UPS) de la ESPAM MFL.

## **2.3. ENTREGABLES DEL PROYECTO**

A continuación, se indican y describen cada uno de los entregables que serán generados y utilizados por el proyecto (Cuadro 2).

Es preciso destacar que de acuerdo a la filosofía de MEIDAW (y de todo proceso iterativo e incremental), todos los entregables son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso se podría tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los entregables.

**Cuadro 2.** Lista de entregables

Número	Entregable	Descripción
1	Plan de Desarrollo de Software	Este documento.
2	Especificación de Requisitos de Software (ERS)	Los requisitos del núcleo, basándose en las directrices dadas por el estándar Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.
3	Diagramas UML: Casos de Uso.	Desde la perspectiva de actores y sistemas externos, detalla el comportamiento del núcleo mediante los diagramas UML.
4	BPMN 2.0.	Contiene los elementos del estándar BPMN 2.0 utilizados por el modelador, además especifica sus características, definición y los atributos utilizados y las posibles formas que pueden conectarse entre sí.
5	Plan de iteración	Un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.
6	Plan de pruebas	Especifica las condiciones de los dos casos de prueba: entradas, condiciones de ejecución y resultados esperados. Al ejecutar las pruebas, se presentaran los resultados obtenidos y se compararán con los resultados esperados. Además, cada caso de prueba lleva asociado un procedimiento de prueba con las instrucciones para realizar la misma.
7	Manual del programador	De forma clara y precisa se detalla la documentación del código, principalmente las clases, objetos, variables, métodos y procedimientos utilizados durante la programación, además del diccionario de datos correspondiente a la BD del sistema.
8	Producto	Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la Fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva versión al final de cada iteración.

**Elaboración:** Autores

En cada entregable, si se considera necesario, se va a destinar un capítulo para el glosario y/o abreviaturas, incluyéndose uno para este documento. Los entregables 4, 5, 6, 7 y 8 se generarán a partir de la fase de Construcción, por lo cual se han incluido aquí sólo para dar una visión global de los mismos.

### 3. ORGANIZACIÓN DEL PROYECTO

#### 3.1. PARTICIPANTES EN EL PROYECTO

**Jefe de Proyecto.** - Ing. Joffre Moreira, docente encargado de tutelar el presente trabajo.

**Programadores.** - Dayana Bailón y José Mera, egresados de la carrera de computación de la ESPAM MFL, autores de este trabajo.

### 3.2. INTERFACES EXTERNAS

Como consultores para la realización del núcleo del Workflow se contará con los siguientes participantes, mismos que interactuarán con los desarrolladores para conocer la especificación de los requerimientos del núcleo: Ing. Javier López, .Ing. Luis Ulloa, Ing. Javier Cedeño, Ing. Gustavo Donoso.

### 3.3. ROLES Y RESPONSABILIDADES

A continuación, se describen las principales responsabilidades de cada uno de los participantes del proyecto (Cuadro 3):

**Cuadro 3.** Roles y responsabilidades.

Cargo	Responsabilidad
<b>Jefe del Proyecto</b>	El jefe de proyecto gestiona las prioridades y mantiene al equipo del proyecto enfocado en los objetivos
<b>Programadores y Administradores de Base de Datos</b>	Construcción de prototipos. Elaboración de las pruebas funcionales, modelo de datos y en las validaciones por defecto.

Elaboración: Autores

## 4. GESTIÓN DEL PROCESO

### 4.1. ESTIMACIONES DEL PROYECTO

El presupuesto de este proyecto, así como los recursos utilizados están detallados en el Cuadro 4.

**Cuadro 4.** Presupuesto de desarrollo.

Actividad	Detalle	Cantidad	Precio unit.	Precio total
Identificar los requerimientos generales del núcleo	Internet	50 horas	\$ 0,50	\$ 25,00
	Laptop Cori5 RAM 8GB	2	\$ 700,00	\$ 1400,00
	Viáticos	10	\$ 2,50	\$ 25,00
Determinar las herramientas de desarrollo.	Internet	50 horas	\$ 0,50	\$ 25,00
	Capacitación sobre desarrollo de Workflow	5 horas	\$ 10,00	\$ 50,00
	Laptop Cori5 RAM 8GB	2	-	-
	Viáticos	10	\$ 2,50	\$ 25,00
Definir el esquema de funcionamiento del núcleo.	Internet	100 horas	\$ 0,50	\$ 50,00
	Laptop Cori5 RAM 8GB	2	-	-
	Viáticos	15	\$ 2,50	\$ 37,50

Definir las actividades, responsables y tiempos, estimados para el desarrollo.	Laptop Cori5 RAM 8GB	-	-	-
	Viáticos	10	\$ 2,50	\$ 25,00
Configurar la arquitectura de desarrollo	Internet	25 horas	\$ 0,50	\$ 12,50
	Laptop Cori5 RAM 8GB	2	-	-
Diseñar e implementar la arquitectura de datos.	Internet	100 horas	\$ 0,50	\$ 50,00
	Capacitación sobre BD y cubos de datos	5 horas	\$ 10,00	\$ 50,00
	Laptop Cori5 RAM 8GB	2	-	-
	Viáticos	20	\$ 2,50	\$ 50,00
Desarrollar la capa intermedia (Negocio) de la plataforma.	Internet	300 horas	\$ 0,50	\$ 150,00
	Laptop Cori5 RAM 8GB	2	-	-
	Capacitación sobre proyectos de Workflow	5 horas	\$ 10,00	\$ 50,00
	Viáticos	10	\$ 2,50	\$ 150,00
Programar notificaciones y escalaciones de la plataforma	Internet	200	\$ 0,50	\$ 100
	Laptop Cori5 RAM 8GB	2	-	-
	Viáticos	20	\$ 2,50	\$ 50,00
Diseño del Plan de Pruebas	Laptop Cori5 RAM 8GB	2	-	-
Ejecución de pruebas funcionales y de carga de datos.	Laptop Cori5 RAM 8GB	2	-	-
	Servidor Intel Xeon -2.53 GHz -RAM 6 Gb. -64 bits - Disco duro de 2 teras 7200 rpm SATA3. -Windows Server 2008 R2.	1	-	-
	Laptop Cori5 RAM 8GB	2	-	-
	Servidor Intel Xeon -2.53 GHz -RAM 6 Gb. -64 bits - Disco duro de 2 teras 7200 rpm SATA3. -Windows Server 2008 R2.	1	-	-
Integrar el núcleo y las interfaces.	Laptop Cori5 RAM 8GB	2	-	-
Instalar el sistema en el servidor.	Servidor Intel Xeon -2.53 GHz -RAM 6 Gb. -64 bits - Disco duro de 2 teras 7200 rpm SATA3. -Windows Server 2008 R2.	1	\$ 2000,00	\$ 2000,00
	Laptop Cori5 RAM 8GB	2	-	-
Verificar la correcta ejecución en el control de flujos de trabajo de los procesos administrativos	Servidor Intel Xeon -2.53 GHz -RAM 6 Gb. -64 bits - Disco duro de 2 teras 7200 rpm SATA3. -Windows Server 2008 R2.	1	-	-
	Laptop Cori5 RAM 8GB	2	-	-
Elaboración del informe de tesis	Laptop Cori5 RAM 8GB	2	-	-
	Imprevistos	1	\$ 415,00	\$ 415,00
			<b>Total</b>	<b>\$ 4.565,00</b>

**Elaboración:** Autores

## 4.2. PLAN DEL PROYECTO

### PLAN DE FASES

De acuerdo a las fases principales de MEIDAW, se establecieron las sub-fases de las mismas con su respectiva duración (Cuadro 5), para alcanzar la ejecución de este proyecto.

**Cuadro 5.** Plan de fases de proyecto.

Fase	Subfase	Duración
<b>ORIENTACIÓN AL PROCESO</b>	Plan del proyecto	1 mes
	Documento de Alcance y Requerimientos	1 mes
	Planificación de Incrementos	-
<b>ORIENTACIÓN AL PRODUCTO</b>	Análisis del Sistema	2 meses
	Diseño detallado del Sistema	1 mes
	Desarrollo del Sistema	8 meses
	Pruebas del Sistema	1 mes
	Documentos del Sistema	1 mes

Elaboración: Autores

### CRONOGRAMA DEL DESARROLLO DEL PROYECTO

Una vez definidas las fases: orientación al proceso y orientación al producto, con su respectiva duración, y con base a los objetivos específicos del proyecto y actividades de los mismos, se determinó el cronograma de desarrollo del núcleo (Cuadro 6), especificando los responsables y tiempos estimados por sub-actividad.

**Cuadro 6.** Cronograma de desarrollo del proyecto.

Actividad	Sub-actividad	Responsables	Duración (días)	Fecha de inicio	Fecha de fin
Configurar la arquitectura.	Instalar Visual Studio 2015.	Dayana Bailón	2	05/12/16	06/12/16
	Instalar SQL SERVER 2008.	José Mera	3	07/12/16	09/12/16
Diseñar e implementar la arquitectura de datos.	Diseñar el esquema de la BD.	Dayana Bailón José Mera	10	12/12/16	23/12/16
	Implementar el esquema en SQL SERVER 2008.	Dayana Bailón José Mera	5	26/12/16	30/12/16
	Desarrollar Stores Procerdures	Dayana Bailón José Mera	5	02/01/17	06/01/17
Desarrollar la capa intermedia (Negocio).	Crear capa de acceso a la BD.	José Mera	1	09/01/17	09/01/17
	Desarrollar métodos CRUD.	Dayana Bailón José Mera	144	10/07/17	28/07/17
	Desarrollar Recursos Web.	Dayana Bailón José Mera	144	10/07/17	28/07/17

Programar notificaciones y escalaciones	Configurar un cron job para verificar el cumplimiento de las tareas.	Dayana Bailón José Mera	20	03/07/17	28/07/17
Diseño del plan de pruebas	Solicitar la documentación de un proceso administrativo.	José Mera	1	01/08/17	02/08/17
	Establecer un proceso, como sujeto de prueba.	Dayana Bailón	1	02/08/17	02/08/17
	Realizar una entrevista al Coordinador de la UPS para conocer los detalles del proceso seleccionado.	Dayana Bailón José Mera	1	03/08/17	03/08/17
	Realizar la documentación del plan de pruebas.	Dayana Bailón José Mera	2	03/08/17	04/08/17
Ejecución de pruebas	Ejecución de pruebas funcionales.	José Mera	5	07/08/17	11/08/17
	Documentar los errores.	Dayana Bailón	5	07/08/17	11/08/17
	Ejecución de prueba de carga de datos.	José Mera	5	14/08/17	18/08/17
	Documentar los errores.	Dayana Bailón	5	14/08/17	18/08/17
Gestión de incidencias	Corregir los errores documentados.	Dayana Bailón José Mera	5	14/08/17	18/08/17
	Realizar el manual del programador y del administrador.	Dayana Bailón José Mera	9	21/08/17	31/08/17
Integrar el Núcleo con las interfaces	Permitir acceso a los recursos.	Dayana Bailón	5	04/09/17	08/09/17
Instalar el sistema en el servidor	Instalar el sistema.	José Mera	5	11/09/17	15/09/17
	Verificar el correcto funcionamiento del sistema.	Dayana Bailón	10	18/09/17	29/09/17

Fuente: Autores

### 4.3. SEGUIMIENTO Y CONTROL DEL PROYECTO

#### GESTIÓN DE REQUISITOS

Los requisitos del sistema son especificados en el Anexo 2 Estándar IEEE 830. Cada requisito tendrá un diagrama de casos de uso (Anexo 3).

#### CONTROL DE PLAZOS

El cronograma del proyecto tendrá un seguimiento semanal por el jefe de proyecto y por los integrantes de la tesis (programadores). Además se llevará el control de los incrementos planificados, cada 15 días (Anexo 5).

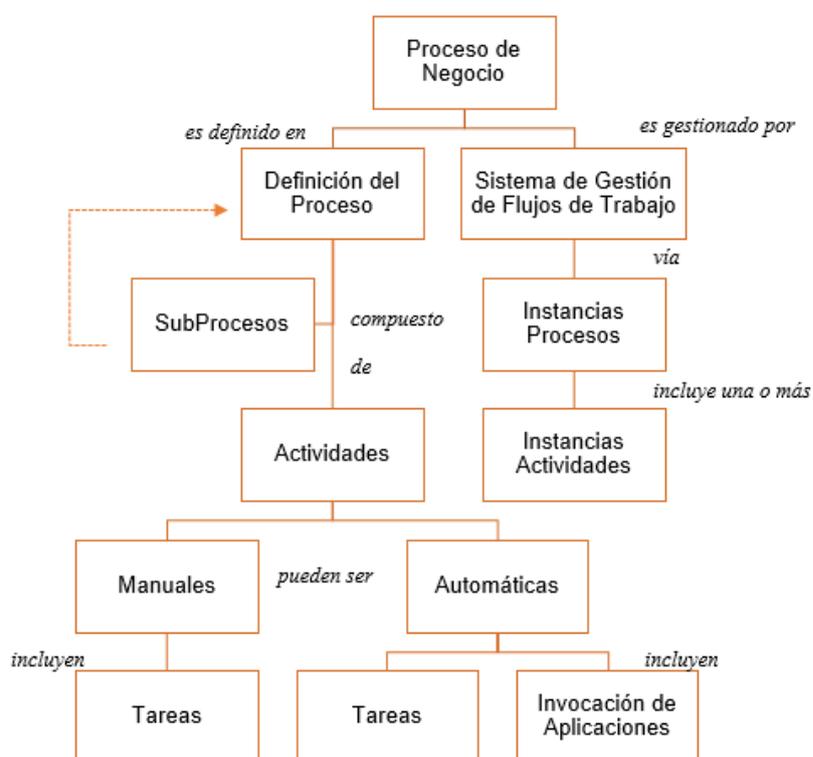
#### CONTROL DE CALIDAD

Los defectos detectados en las revisiones tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias.

## 5. ESPECIFICACIONES DEL PRODUCTO

### 5.1. ESQUEMA DE FUNCIONAMIENTO DEL NÚCLEO.

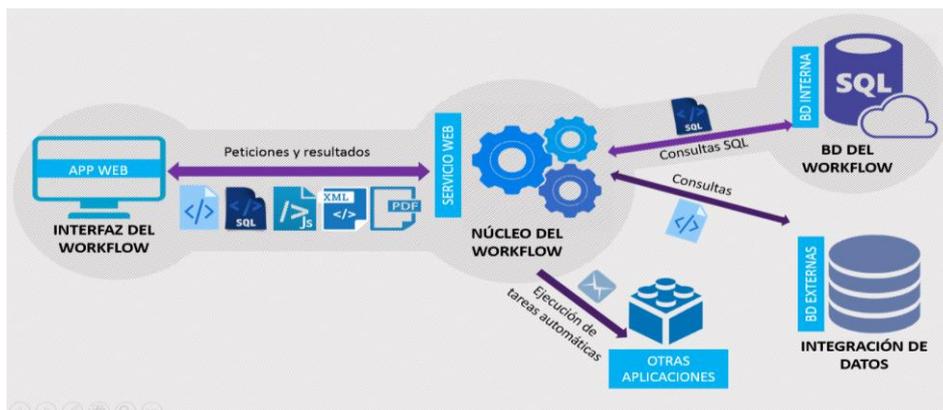
Mediante el método analítico, se analizaron los diversos elementos que pueden intervenir en los procesos administrativos (Figura 1); comprendiendo el concepto y función de las tareas, actividades y participantes, además de las relaciones entre sí, y las respectivas instancias de los procesos.



**Figura 1.** Esquema general de la gestión de los procesos administrativos.

**Elaboración:** Autores

Luego de dicho análisis, y de acuerdo a lo recomendado por la WfMC (Workflow Management Coalition), se define el esquema presentado en la Figura 2, mismo que especifica la comunicación entre el motor y los demás elementos que intervienen en el Workflow; las peticiones del usuario mediante las interfaces, las consultas de datos a la BD interna de la herramienta y a las externas (institucionales); y las interacciones con aplicaciones de terceros.



**Figura 2.** Esquema de funcionamiento general del núcleo.  
Fuente y Elaboración: Autores

La WfMC ha recomendado un conjunto de interfaces que deben cumplir los sistemas Workflow, por lo cual es necesario que el núcleo pueda dar soporte a las mismas, a fin de que pueda proveer el control del ambiente de ejecución de las instancias de los flujos de trabajo. Es por esto, que se utilizaron dichas interfaces como referencia, detallándose en el Cuadro 7 los requerimientos generales que debe alcanzar el núcleo.

**Cuadro 7.** Requerimientos generales del núcleo.

Identificación	Nombre	Característica
RG01	Herramientas de definición de procesos.	Para el manejo de la gestión de los procesos administrativos y sus respectivas versiones.
RG02	Aplicaciones clientes	Para la ejecución de las tareas y actividades de los flujos de trabajo.
RG03	Aplicaciones Invocadas	Para levantar aplicaciones de terceros, que contribuyan en el desarrollo del flujo de trabajo.
RG04	Herramientas de administración y monitoreo.	Para verificar el cumplimiento de las tareas de los flujos de trabajo.

Fuente: Autores.

## 5.2. PILA PARA EL DESARROLLO LENGUAJE DE PROGRAMACIÓN

TIOBE (2016) ha establecido un ranking (top 50) de los lenguajes de programación más populares. El índice se elabora a partir de diversas variables, como por ejemplo, número de ingenieros cualificados en determinado lenguaje, las búsquedas que hacen los usuarios por información de los distintos lenguajes de programación, la demanda de cursos, o los

lenguajes que están siendo más utilizados. Cabe destacar que el índice, actualizado mensualmente, no dice cuál es el mejor, o qué lenguaje de programación es con el que se han escrito más cantidad de líneas de código. Los datos especificados en el Cuadro 8 corresponden al mes de octubre de 2016.

**Cuadro 8.** Los cinco lenguajes de programación más populares.

N.	LENGUAJE DE PROGRAMACIÓN	PUNTUACIONES
1	Java	18.75%
2	C	9.20%
3	C++	5.41%
4	C#	3.65%
5	Python	3.56%

Fuente: TIOBE (2016)

A continuación, a fin de seleccionar el lenguaje de programación más adecuado para el desarrollo del núcleo, de los presentados por TIOBE (2016), se detalla la comparación (Cuadro 9) realizada por Stein & Geyer-Schulz (2013) de las características de Java, Python y C#, este último ha sido seleccionado en contraste con C y C++ por ser una evolución de ellos, destacándose por su sencillez y modernidad.

**Cuadro 9.** Tabla comparativa entre Java, C# y Python.

LENGUAJE	ESTRATEGIA DE EJECUCIÓN	TIPADO DE SISTEMA	PARADIGMA DE PROGRAMACIÓN	GENÉRICOS
JAVA	<b>JIT (Just-in-time) Compilado:</b> traduce código fuente en una representación intermedia de bytecode que se compila en código de máquina especializada en el tiempo de carga o durante la ejecución. En este último caso, el entorno normalmente identifica rutas de código frecuentes ("calientes") y genera código de máquina optimizado para esas partes. Así, la velocidad de ejecución mejora mientras el programa se está ejecutando.	<b>Estático:</b> la comprobación de tipificación se realiza durante la compilación, y no durante la ejecución. Comparado con el tipado dinámico, el estático permite que los errores de programación sean detectados antes, y que la ejecución del programa sea más eficiente.	<b>POO (Programación Orientada a Objetos):</b> es una forma especial de programar, más cercana a la forma en que se expresarían los objetos en la vida real. Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.	<b>E-erasure:</b> Cuando compila algún código contra un tipo o método genérico, el compilador calcula lo que realmente quiere decir (es decir, el argumento de tipo para T) y verifica en tiempo de compilación que está haciendo lo correcto, el compilador genera modelos adicionales cuando sea necesario. En el momento de la ejecución, una Lista <String> y una Lista <Fecha> son exactamente iguales; la información de tipo extra ha sido borrada por el compilador.

<b>C#</b>	<p><b>JIT (Just-in-time) Compilado:</b> al igual que Java el código es compilado a tiempo y mediante técnicas de optimización. C# implementa esta técnica mediante Microsoft Common Language Runtime (CLR).</p>	<p><b>Estático y Dinámico combinados:</b> utilizan declaraciones de tipos de datos dinámicos en un ambiente estático, permitiendo la flexibilidad casi completa de un lenguaje dinámico en un lenguaje estático.</p>	<p><b>POO (Programación Orientada a Objetos):</b> al igual que Java permite que el programador pueda establecer elementos como si fuesen objetos del mundo real.</p>	<p><b>Runtime:</b> El Entorno en tiempo de Ejecución de Lenguaje Común (CRL) es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada Common Intermediate Language (CIL) al código de máquina nativo, mediante un compilador en tiempo de ejecución.</p>
<b>PYTHON</b>	<p><b>Intérprete:</b> ejecuta el código paso a paso. Cada instrucción se analiza y sobre la marcha se traduce en código de máquina que afecta negativamente el tiempo de ejecución. Python implementa CPython, que incluye un intérprete de bytecode escrito en C y en lugar de operar directamente en el código fuente, es el bytecode el que tiene que interpretarse.</p>	<p><b>Dinámico:</b> la comprobación de tipificación se realiza durante su ejecución en vez de durante la compilación. Comparado con el tipado estático, el dinámico es más flexible (debido a las limitaciones teóricas de la decidibilidad de ciertos problemas de análisis de programas estáticos, que impiden el mismo nivel de flexibilidad que se consigue con el tipado estático), a pesar de ejecutarse más lentamente y ser más propensos a contener errores de programación.</p>	<p>POO (Programación Orientada a Objetos) y Programación Funcional: Contiene varios aspectos de ambos: la orientada a objetos y la funcional, esta última se basa en el concepto de función (que no es más que una evolución de los predicados), de corte más matemático.</p>	<p>No aplica: Python se escribe de forma dinámica, por lo que el concepto de genéricos estáticamente seguros de tipo no se aplica.</p>

**Fuente:** Stein & Geyer-Schulz (2013)

El objetivo del estudio realizado por Stein & Geyer-Schulz (2013) fue examinar cinco lenguajes de programación (C++, Java, C#, F#, Python y su variante Cython) en un escenario computacionalmente intensivo y determinar el rendimiento en tiempo de ejecución, el consumo de memoria y el tamaño del código. En la conclusión, luego de las pruebas aplicadas, indicaron que C# es uno de los lenguajes con tiempo de ejecución más rápidos, a la par de C++ y Java. Y como se mencionó anteriormente, C# es una evolución de C++, mismo

que mejora las características más destacables del lenguaje Java (tal como se especifica en el cuadro 8), por lo cual ha sido selecto para el desarrollo del núcleo del Workflow.

**VISUAL C#:** Es en resumen, un lenguaje de programación que se destaca por su robustez y sencillez, basado en el paradigma POO y que combina el tipado de sistema estático junto con el dinámico, utilizando como estrategia de ejecución JIT-Compiled, mediante su CLR, destinado para compilar CIL al código de maquina nativo.

### **ENTORNO DE DESARROLLO INTEGRADO (IDE)**

En la 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (2016) se especifica que los Entornos de Desarrollo Integrados son muy populares entre los desarrolladores de software, ya que proporcionan soporte para muchas de sus tareas diarias de desarrollo o mantenimiento, siendo los IDE modernos, como Visual Studio (VS), los que proveen depuradores integrados, refactorizaciones automatizadas, herramientas de asistencia como finalización de código e incluso control de versión integrado.

**VISUAL STUDIO (VS):** Microsoft (2011) citado por Arbeláez *et al.* (2011) en su sitio oficial presenta a VS como su IDE, el cual soporta varios lenguajes de programación, entre ellos el seleccionado para el desarrollo del núcleo: Visual C#, permitiendo la creación de aplicaciones de escritorio, aplicaciones de tipo cliente servidor, aplicaciones y servicios Web, entre otras.

### **FRAMEWORK**

De acuerdo a Orellana (2013) un framework facilita el desarrollo de software permitiendo a los diseñadores y programadores dedicar su tiempo a lograr los requerimientos de software en lugar de lidiar con los detalles de bajo nivel necesarios para obtener un sistema funcional. Microsoft (2012) dispone de .NET Framework, una tecnología que soporta la compilación y ejecución de aplicaciones y servicios Web XML de última generación.

Por lo cual, según lo especifican Arbeláez *et al.* (2011) se desarrolla ASP.NET, una plataforma web compilada en .NET Framework que proporciona todos los servicios necesarios para compilar aplicaciones web empresariales basadas en servidor, mediante distintos lenguajes de programación, entre ellos C#. Ceballos (2012) destaca que la plataforma ASP.NET es un modelo diseñado para proporcionar un entorno de ejecución distribuido y multicapa (interfaz de usuario, servicios de aplicación y lógica de negocio, y servicios de gestión de datos).

**ASP.NET WEBAPI:** es el nuevo framework de la familia .NET, que tiene como objetivo facilitar en gran medida la construcción de APIs RESTful orientadas a ofrecer servicios (Bezanila, 2015), por lo que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles (Microsoft, 2015).

#### **SISTEMA GESTOR DE BASE DE DATOS (SGBD)**

Martín *et al.* (2016) especifican que un SGBD permite crear, modificar y mantener una base de datos, siendo capaz de poner a disposición de muchos usuarios grandes cantidades de datos, de manera simultánea. Por su parte, Fotache y Strimbei (2015) indican que entre las bases de datos SQL y las noSQL, SQL demuestra una competencia de base invaluable, extrayendo conjuntos de registros de enormes bases de datos basadas en álgebra relacional.

Estos últimos también detallan que la inmensa popularidad de SQL (Michael Stonebraker, científico especializado en la base de datos de investigación y desarrollo, una vez lo llamó el lenguaje intergaláctico de datos) se debe principalmente a su sintaxis de alto nivel (la programación no es necesaria para la mayoría de las consultas) y también a su implementación en todos los tipos SGBD, desde el escritorio (Access) a open-source (MySQL, PostgreSQL) y comerciales (Oracle Database, Microsoft SQL Servers). La amplia adopción fue

facilitada por la estandarización de SQL por ISO con ANSI y varias agencias nacionales.

**MICROSOFT SQL SERVER:** Definido por Martín *et al.* (2016) como un SGBD basado en el lenguaje Transact-SQL, que según Fotache y Strimbei (2015) es capaz de analizar grandes conjuntos de tablas relacionales y el procesamiento de modelos estadísticos.

## 6. GLOSARIO Y ABREVIATURAS

### 6.1. GLOSARIO

- **.NET Framework:** plataforma de desarrollo y ejecución de aplicaciones, propio de Microsoft.
- **Access:** SGBD relacional creado y modificado por Microsoft.
- **API RestFul:** forma de nombrar a aquellas APIs que si cumplen de forma correcta con la definición de REST.
- **ASP.NET:** marco web de código abierto para la creación de aplicaciones y servicios web modernos con .NET.
- **C#:** (pronunciado C Sharp) es un lenguaje de programación bajo el paradigma POO, desarrollado por Microsoft, derivado de C/C++ y Java.
- **C:** lenguaje de programación de medio nivel.
- **C++:** lenguaje de programación multiparadigma, derivado de C.
- **Entregable:** documento medible y verificable que se elabora para completar un proyecto o parte de un proyecto, destinado a un miembro interno o externo del equipo de desarrollo.
- **F#:** (pronunciado F Sharp) lenguaje de programación multiparadigma de código abierto de Microsoft.
- **Framework:** conjunto de componentes, herramientas y documentación que facilita el desarrollo de software.
- **Java:** lenguaje de programación de propósito general administrada por Oracle.

- **Microsoft:** empresa estadounidense dedicada al sector de software y el hardware.
- **NoSQL:** Base de datos no relacionales.
- **Núcleo:** software que se encarga del seguimiento de los casos o instancias de los procesos implementados en el Workflow.
- **Oracle Corporation:** compañía de software que desarrolla bases de datos y sistemas de gestión de bases de datos.
- **Oracle Database:** SGBD de tipo objeto-relacional desarrollado por Oracle Corporation.
- **Proceso iterativo e incremental:** analiza el impacto de cada nuevo cambio del proyecto para no destruir lo anterior completamente, sino modificarlo ligeramente para extenderlo al mínimo costo.
- **Python:** lenguaje de programación multiparadigma administrado por la Fundación de Software Python.
- **REST:** técnica de arquitectura de software, es decir, un conjunto de principios y patrones de comunicación que ayudan a crear una forma de pensar y construir las APIs.
- **Sybase:** fue una compañía dedicada al desarrollo de tecnología de la información.
- **Transact-SQL:** es una extensión al SQL de Microsoft y Sybase.
- **Workflow:** automatización de un proceso de negocio, en todo o en parte, durante el cual los Documentos, Información o las Tareas (Actividades) se pasan de un participante a otro para que realicen alguna acción, de acuerdo a un conjunto de reglas procedimentales.

## 6.2. ABREVIATURAS

- **ANSI:** American National Standards Institute.
- **API:** Application Programming Interface.
- **BPMN:** Business Process Modeling Notation
- **CIL:** Common Intermediate Language.
- **CLR:** Common Language Runtime.
- **ERS:** Especificaciones de Requisitos de Software.

- **ESPAM MFL:** Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.
- **HTTP:** Hypertext Transfer Protocol.
- **IDE:** Integrated Development Environment.
- **IEEE:** Institute of Electrical and Electronics Engineers.
- **ISO:** International Organization for Standardization.
- **MEIDAW:** Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow.
- **POO:** Programación Orientada a Objetos.
- **RAM:** Random Access Memory.
- **REST:** Representational State Transfer.
- **SANER:** Software Analysis, Evolution, and Reengineering.
- **SGBD:** Sistema Gestor de Base de Datos.
- **SQL:** Structured Query Language.
- **TIOBE:** The Importance Of Being Earnest.
- **UML:** Unified Modeling Language.
- **UPS:** Unidad de Producción de Software.
- **VS:** Visual Studio.
- **XML:** Extensible Markup Language.

## 7. REFERENCIAS

- 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER). (2016, Osaka, Japón). 2016. A Study of Visual Studio Usage in Practice. DE. p 1-3.
- Arbeláez, O; Medina, F; Chaves, O. 2011. Herramientas para el desarrollo rápido de aplicaciones web. Pereira, CO. Scientia Et Technica. Vol. 17. p 254 - 258.
- Bezanila, P. 2015. Gestión de datos para aplicaciones generales usando tecnologías .NET y metodologías de tipo 'Design Pattern'. Tesis. Ing. de Telecomunicación. Universidad de Cantabria. ES. p 14.
- Fotache, M. y Strimbei, C. 2015. SQL and data analysis. Some implications for data analysis and higher education. RO. Journal Procedia Economics and Finance. Vol. 20, p. 243-251.

- Martín, M; Leyva, G; Peña, N; Labrada, R. 2016. Proceso de réplica de datos con Microsoft SQL Server para el Replicador de Datos Reko. CU. Revista Cubana de Ciencias Informáticas. Vol. 10. p 128-142.
- Microsoft. 2012. Información general acerca de .NET Framework. (En línea). Consultado, 12 de Oct. 2016. Formato HTML. Disponible en <https://msdn.microsoft.com>
- Microsoft. 2015. ASP.NET Web API. (En línea). Consultado, 12 de Oct. 2016. Formato HTML. Disponible en <https://msdn.microsoft.com>
- Orellana, M. 2013. Evaluación de frameworks realizado en java para aplicaciones on-line. Tesis. Ing. en Sistemas. Universidad Nacional Autónoma de México. MX. p 3.
- Stein, M. & Geyer-Schulz, A. 2013. A Comparison of Five Programming Languages in a Graph Clustering Scenario. Karlsruhe, DE. Revista Universal Computer Science. Vol. 19. p 428-456.
- TIOBE. 2016. TIOBE Index for October 2016. (En línea). Consultado, 10 de Oct. 2016. Formato HTML. Disponible en <http://www.tiobe.com/tiobe-index/>

## ANEXO 2 – ESPECIFICACIÓN DE REQUERIMIENTOS



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

# IEEE830

## Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL



**ESPECIFICACIONES  
DE REQUISITOS  
DE SOFTWARE**

Dayana H. Bailón Delgado  
José S. Mera Cantos

**2017**

## FICHA DEL DOCUMENTO

El cuadro 1 detalla las versiones que ha tenido este anexo, indicando: fecha, versión, autores y por quiénes fue revisado.

**Cuadro 10.** Ficha del documento.

<b>Fecha</b>	<b>Versión</b>	<b>Autores</b>	<b>Verificado por</b>
<b>10/10/2016</b>	1.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Javier López
<b>20/10/2016</b>	2.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira
<b>29/11/2016</b>	3.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira
<b>10/08/2017</b>	4.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira

## CONTENIDO GENERAL

FICHA DEL DOCUMENTO .....	138
CONTENIDO GENERAL.....	139
CONTENIDO DE CUADROS .....	140
INTRODUCCIÓN .....	142
1.1 PROPÓSITO .....	142
1.2 ALCANCE.....	142
1.3 PERSONAL INVOLUCRADO.....	142
1.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS .....	143
1.5 REFERENCIAS.....	144
1.6 RESUMEN.....	144
DESCRIPCIÓN GENERAL .....	144
1.1 PERSPECTIVA DEL PRODUCTO .....	144
1.2 CARACTERÍSTICAS DE LOS USUARIOS .....	144
1.3 RESTRICCIONES .....	145
1.4 SUPOSICIONES Y DEPENDENCIAS.....	146
REQUISITOS ESPECÍFICOS .....	146
1.5 REQUERIMIENTOS FUNCIONALES.....	146
1.5.1 PROCESOS.....	146
1.5.2 ACTIVIDADES .....	156
1.5.3 PARTICIPANTES DE LA ACTIVIDAD .....	158
1.5.4 CONTROL .....	159
1.5.5 REPORTES .....	161
1.5.6 INTEGRACIONES .....	162
1.6 REQUERIMIENTOS NO FUNCIONALES.....	162
1.6.1 ATRIBUTOS DE CALIDAD .....	163
1.7 REQUISITOS COMUNES DE LAS INTERFACES.....	164
1.7.1 INTERFACES DE HARDWARE.....	164
1.7.2 INTERFACES DE SOFTWARE .....	165
1.7.3 INTERFACES DE COMUNICACIÓN.....	165

## CONTENIDO DE CUADROS

<b>Cuadro 1.</b> Ficha del documento.....	138
<b>Cuadro 2.</b> Datos del personal involucrado 1.....	143
<b>Cuadro 3.</b> Datos del personal involucrado 2.....	143
<b>Cuadro 4.</b> Datos del personal involucrado 3.....	143
<b>Cuadro 5.</b> Definiciones, acrónimos y abreviaturas. ....	143
<b>Cuadro 6.</b> Referencias.....	144
<b>Cuadro 7.</b> Características del Administrador. ....	145
<b>Cuadro 8.</b> Características del Supervisor. ....	145
<b>Cuadro 9.</b> Características del Operador. ....	145
<b>Cuadro 10.</b> Características del Modelador de diagramas.....	145
<b>Cuadro 11.</b> Características del Diseñador.....	145
<b>Cuadro 12.</b> Características del Administrador de Base de Datos (DBA). ....	145
<b>Cuadro 13.</b> Requerimiento Funcional 01. ....	148
<b>Cuadro 14.</b> Requerimiento Funcional 02. ....	149
<b>Cuadro 15.</b> Requerimiento Funcional 03. ....	149
<b>Cuadro 16.</b> Requerimiento Funcional 04. ....	150
<b>Cuadro 17.</b> Requerimiento Funcional 05. ....	150
<b>Cuadro 18.</b> Requerimiento Funcional 06. ....	151
<b>Cuadro 19.</b> Requerimiento Funcional 07. ....	151
<b>Cuadro 20.</b> Requerimiento Funcional 08. ....	152
<b>Cuadro 21.</b> Requerimiento Funcional 09. ....	152
<b>Cuadro 22.</b> Requerimiento Funcional 10. ....	153
<b>Cuadro 23.</b> Requerimiento Funcional 11. ....	153
<b>Cuadro 24.</b> Requerimiento Funcional 12. ....	154
<b>Cuadro 25.</b> Requerimiento Funcional 13. ....	154
<b>Cuadro 26.</b> Requerimiento Funcional 14. ....	155
<b>Cuadro 27.</b> Requerimiento Funcional 15. ....	155
<b>Cuadro 28.</b> Requerimiento Funcional 16. ....	156
<b>Cuadro 29.</b> Requerimiento Funcional 17. ....	157
<b>Cuadro 30.</b> Requerimiento Funcional 18. ....	157
<b>Cuadro 31.</b> Requerimiento Funcional 19. ....	158
<b>Cuadro 32.</b> Requerimiento Funcional 20. ....	158

<b>Cuadro 33.</b> Requerimiento Funcional 21.....	159
<b>Cuadro 34.</b> Requerimiento Funcional 22.....	159
<b>Cuadro 35.</b> Requerimiento Funcional 23.....	160
<b>Cuadro 36.</b> Requerimiento Funcional 24.....	160
<b>Cuadro 37.</b> Requerimiento Funcional 25.....	160
<b>Cuadro 38.</b> Requerimiento Funcional 26.....	161
<b>Cuadro 39.</b> Requerimiento Funcional 27.....	162

## **INTRODUCCIÓN**

Este documento es una Especificación de Requisitos Software (ERS) para el núcleo de un Workflow de procesos administrativos en la ESPAM MFL, se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

### **PROPÓSITO**

Mediante el documento, se detallan los requerimientos generales que deberá cumplir el núcleo del Workflow. El objetivo del core es procesar la información de las actividades que intervienen en los diferentes procesos administrativos de la ESPAM MFL para asignarlas a los recursos correspondientes, ya sean humanos o tecnológicos, a fin de que cumplan con las respectivas tareas: manuales o automáticas.

Este documento está dirigido a los encargados de dar el mantenimiento necesario del núcleo del Workflow, mismo que permitirá dar a conocer todos los requerimientos funcionales y no funcionales del core de la plataforma.

### **ALCANCE**

El desarrollo del núcleo del workflow se realizará con base en los requerimientos generales especificados en el Plan de Proyecto (Anexo 1), y en los requerimientos de la presente ERS.

### **PERSONAL INVOLUCRADO**

Los usuarios que se describen a continuación (Cuadros 2, 3 y 4) son aquellos que están involucrados de manera directa con el desarrollo del proyecto.

**Cuadro 11.** Datos del personal involucrado 1.

<b>Nombre</b>	<b>Dayana Bailón</b>
<b>Rol</b>	Analista, DBA y programadora.
<b>Responsabilidad</b>	Análisis de información, diseño y programación del núcleo del Workflow
<b>Información de contacto</b>	dayu_espam@outlook.es

Elaboración: Autores

**Cuadro 12.** Datos del personal involucrado 2.

<b>Nombre</b>	<b>José Mera</b>
<b>Rol</b>	Analista, DBA y programador
<b>Responsabilidad</b>	Análisis de información, diseño y programación del núcleo del Workflow
<b>Información de contacto</b>	josmer1202@hotmail.com

Elaboración: Autores

**Cuadro 13.** Datos del personal involucrado 3.

<b>Nombre</b>	<b>Joffre Moreira</b>
<b>Rol</b>	Tutor
<b>Responsabilidad</b>	Tutelar el desarrollo del núcleo, verificando el cumplimiento de las especificaciones estipuladas en el documento.
<b>Información de contacto</b>	jmoreira@espam.edu.ec

Elaboración: Autores

## DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

En el cuadro 5 se especifican las abreviaturas utilizadas en el desarrollo de este documento.

**Cuadro 14.** Definiciones, acrónimos y abreviaturas.

<b>Nombre</b>	<b>Descripción</b>
<b>ANSI</b>	American National Standards Institute.
<b>API</b>	Application Programming Interface
<b>BPMN</b>	Business Process Manager Notation
<b>ERS</b>	Especificación de Requisitos Software
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IEEE</b>	Institute of Electrical and Electronic Engineers
<b>REST</b>	Representational State Transfer
<b>RF</b>	Requerimiento Funcional
<b>RNF</b>	Requerimiento No Funcional
<b>SGFT</b>	Sistema de Gestión de Flujo de Trabajo
<b>Sql Server Agent</b>	Servicio de Microsoft que ejecuta tareas administrativas programadas, denominadas trabajos (jobs).
<b>Usuario</b>	Persona que usará el sistema para gestionar procesos
<b>WfMC</b>	Workflow Management Coalition

Elaboración: Autores

## REFERENCIAS

Las referencias o citas utilizadas para realizar la presente ERS, se especifican en el cuadro 6.

Cuadro 15. Referencias.

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE
The Workflow Reference Model	WfMC

Elaboración: Autores

## RESUMEN

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo, además del propósito y alcance del núcleo del workflow.

En la segunda sección del documento se realiza una descripción general del núcleo, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos funcionales y no funcionales que debe satisfacer el desarrollo del núcleo.

## DESCRIPCIÓN GENERAL

### 1.1 PERSPECTIVA DEL PRODUCTO

El núcleo del Workflow será desarrollado de manera independiente con Base de Datos (BD) propia y será ejecutada por medio de recursos web.

### 1.2 CARACTERÍSTICAS DE LOS USUARIOS

En esta parte se describen los roles de usuario y las actividades que pueden realizar dentro de la aplicación Workflow (Cuadros 7 -12).

**Cuadro 16.** Características del Administrador.

<b>Tipo de usuario</b>	<b>Administrador</b>
<b>Actividades</b>	Puede aprobar o no un nuevo proceso, revisar el cumplimiento de todos los trámites que intervienen en el proceso que administra, reasignar responsables y realizar reportes de dichos procesos.
<b>Elaboración:</b> Autores	

**Cuadro 17.** Características del Supervisor.

<b>Tipo de usuario</b>	<b>Supervisor</b>
<b>Actividades</b>	Puede verificar el cumplimiento de todos los trámites del proceso, reasignar responsables en las actividades y realizar reportes de los procesos.
<b>Elaboración:</b> Autores	

**Cuadro 18.** Características del Operador.

<b>Tipo de usuario</b>	<b>Operador</b>
<b>Actividades</b>	Realiza únicamente los trámites (tareas) a los que ha sido asignado. En caso de que en la etapa de diseño del proceso, durante la asignación de recursos, haya sido definido como usuario responsable del inicio del proceso, puede instanciarlo cuando sea necesario.
<b>Elaboración:</b> Autores	

**Cuadro 19.** Características del Modelador de diagramas.

<b>Tipo de usuario</b>	<b>Modelador</b>
<b>Actividades</b>	Se encarga de modelar el diagrama (flujo) del proceso.
<b>Elaboración:</b> Autores	

**Cuadro 20.** Características del Diseñador

<b>Tipo de usuario</b>	<b>Diseñador</b>
<b>Actividades</b>	Es el responsable de diseñar los formularios que posteriormente permitirán que el Operador registre la respectiva información, además puede generar Vistas de datos y Reportes en PDF que presenten información relevante de los formularios, añadiéndole diseño personalizado.
<b>Elaboración:</b> Autores	

**Cuadro 21.** Características del Administrador de Base de Datos (DBA).

<b>Tipo de usuario</b>	<b>DBA (Administrador de Base de Datos)</b>
<b>Actividades</b>	Puede realizar integraciones externas mediante la ejecución de servicios web, o gestionar integraciones internas para consultar información propia del sistema.
<b>Elaboración:</b> Autores	

### 1.3 RESTRICCIONES

- Lenguajes y tecnologías en uso: ASP.NET, C#, VISUAL STUDIO 2015.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema se diseñará según un modelo cliente/servidor.

## 1.4 SUPOSICIONES Y DEPENDENCIAS

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados, para garantizar una ejecución correcta y la disponibilidad del mismo.

## REQUISITOS ESPECÍFICOS

### 1.5 REQUERIMIENTOS FUNCIONALES

#### 1.5.1 PROCESOS

Los procesos administrativos de la ESPAM MFL están formados por un conjunto de actividades, ordenadas de forma lógica, a fin de alcanzar algún objetivo previamente establecido. Cuando un proceso es ejecutado, desde la plataforma, los responsables de cada actividad, deberán completar las tareas predefinidas, para alcanzar con éxito la finalización del proceso. El núcleo será el encargado de gestionar los siguientes elementos, en los procesos:

**Escalaciones:** Las escalaciones son las vías de acceso que conectan varias actividades de flujo de trabajo, representarán un flujo (continuidad), unidas en un grafo que brinde las relaciones de precedencia del proceso y las transiciones posibles entre actividades. Permitiendo que el Supervisor pueda verificar el estado actual y progreso alcanzado durante la ejecución del proceso y en caso de posibles errores, sea capaz de visualizar el punto concreto del grafo donde ocurrió el problema.

**Parámetros de entrada/salida:** Los parámetros de entrada y salida definirán los datos que se transfieren al proceso de flujo de trabajo y que se devuelven de este. En la definición de un proceso se debe tener en cuenta dichos parámetros, en cada tarea.

El usuario encargado de definir el flujo de trabajo, especificará cuáles son los parámetros mínimos que deberá tener una actividad antes de ser ejecutada.

También se podrán especificar los datos que se requieren como resultado de una actividad, por ejemplo información no estructurada (documentos adjuntos).

Así mismo, existirán actividades (por ejemplo, las tareas manuales) que no requerirán de alguno de estos parámetros, es decir, podrán ser realizadas sin recibir parámetros específicos o sin devolver documentos o datos, luego de ejecutarse la actividad.

**Los roles propios del proceso:** Aquellos roles que deben ser nominados en tiempo de inicio o de ejecución del workflow y que no pueden asignarse de ante mano a un miembro o grupo de la organización.

**Información no estructurada:** El manejo de documentos adjuntos.

**Reglas:** Existirán un conjunto de reglas de negocio, asociadas a los procesos, que no podrán representarse en el grafo debido a su naturaleza o a que tienen un alcance particular de cada actividad. Por lo tanto, estarán incluidos el concepto de validaciones y validaciones específicas.

- ✓ Las validaciones declarativas, incluyen las validaciones globales del proceso (o de parte del proceso) que pueden ser introducidas al diseño como una secuencia de instrucciones IF-THEN-ELSE y que mediante la interpretación del motor de workflow garantizarán la consistencia de la información provista por el usuario o por sistemas externos al completar una actividad.
- ✓ Se dejarán como puntos de entrada para que usuarios con conocimientos apropiados sobre funcionalidades y comportamientos del producto, puedan anexar a las compuertas validaciones específicas que dominen el comportamiento del proceso en el sistema.

Además, el núcleo no solo atenderá las actividades de un proceso, sino que podrán ejecutarse tareas de subprocesos.

**Subprocesos:** Se aplicará el concepto de paquete, definido tal como lo hace la WfMC, considerando al paquete como un contenedor de diseños de procesos, que a su vez puede contener otros paquetes y además se ha dado al sistema la capacidad de definir datos relevantes para la toma de decisiones del proceso, roles, validaciones y eventos a nivel del paquete, aplicando a todos los procesos que se encuentran por debajo de este paquete en la jerarquía.

A continuación, se detallan los Requerimientos Funcionales, que contendrán las descripciones específicas del comportamiento básico del núcleo (Cuadros 13 - 28).

**Cuadro 22.** Requerimiento Funcional 01.

<b>Identificación del requerimiento:</b>	RF01
<b>Nombre del Requerimiento:</b>	Definición de tareas para la creación de un proceso.
<b>Requerimiento Dependiente:</b>	Ninguno.
<b>Características:</b>	Envío de tareas (formulario, diagrama, etc.) necesarias para la definición de un proceso.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ El Administrador del sistema, enviará tareas a los diferentes usuarios, indicando lo que necesita que se realice para la creación de un nuevo proceso.</li> <li>✓ El core se encargará de capturar los datos, guardando las tareas en la bd y enviando un mensaje a cada usuario asignado, indicándole la tarea y el plazo que tiene para realizarla.</li> <li>✓ Una vez que el usuario asignado realiza la tarea, esta cambia a estado ejecutado, y el progreso del conjunto de tareas aumenta.</li> <li>✓ Cuando el progreso este al 100%, automáticamente el proceso pasará a estar listo para la posterior Asignación de Recursos.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 23.** Requerimiento Funcional 02.

<b>Identificación del requerimiento:</b>	<b>RF02</b>
<b>Nombre del Requerimiento:</b>	Supervisión de tareas para diseño del proceso.
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	El usuario Administrador podrá inspeccionar el estado de las tareas de los diversos procesos que estén a su cargo.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Las diferentes tareas a realizarse para la definición de un proceso, deberán ser atendidas, a tiempo, por los respectivos responsables.</li> <li>✓ El Usuario Administrador podrá verificar el cumplimiento de todas las tareas del proceso que haya creado.</li> <li>✓ Si una tarea no es completada a tiempo, se notificará al Administrador sobre lo sucedido, informándose el título y responsable de la misma.</li> <li>✓ El Administrador podrá modificar el plazo de entrega de las tareas y también de usuario responsable.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 24.** Requerimiento Funcional 03.

<b>Identificación del requerimiento:</b>	<b>RF03</b>
<b>Nombre del Requerimiento:</b>	Definición del Flujo (diagrama).
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	Diseño del diagrama del proceso.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Durante el diseño del proceso, el Administrador deberá enviar como tarea al Modelador, el respectivo diagrama del proceso, basándose en las directrices del estándar BPMN 2.0 (en el Anexo 4 se detallan los elementos y reglas del estándar utilizados en el sistema), mismo que será almacenado en la BD para su posterior uso.</li> <li>✓ El flujo del proceso, realizado por el Modelador, será enviado al Administrador para su posterior aprobación.</li> <li>✓ En caso de que el diagrama sea aprobado, será guardado en la BD para su posterior utilización, durante la asignación de recursos y para la ejecución del proceso.</li> <li>✓ En caso de que el diagrama sea rechazado, el Administrador podrá indicar el motivo por el cual tomó esa decisión y el Modelador deberá seguir esas observaciones y volver a realizar la tarea.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 25. Requerimiento Funcional 04.

<b>Identificación del requerimiento:</b>	<b>RF04</b>
<b>Nombre del Requerimiento:</b>	Diseñador de Formularios
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	Se permite al usuario Diseñador crear los formularios que se requieran para que el Operador pueda ingresar la información pertinente, durante la ejecución de un proceso.
<b>Descripción del requerimiento:</b>	<p>Si el usuario Administrador, durante el diseño o rediseño de un proceso, decide que se necesita algún formulario, enviará la tarea al Diseñador, indicando los datos que se deben capturar de dicho formulario.</p> <p>El Diseñador recibe el mensaje y realiza el formulario requerido, la cual es almacenada en la base de datos para su posterior utilización.</p> <p>Cuando el formulario esté listo, se notifica al Administrador el cumplimiento de la actividad.</p> <p>En caso de que el formulario sea aprobado, será guardado en la BD para su posterior utilización, durante la ejecución del proceso.</p> <p>En caso de que el diagrama sea rechazado, el Administrador podrá indicar el motivo por el cual tomó esa decisión y el Diseñador deberá seguir esas observaciones y volver a realizar la tarea.</p>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 26. Requerimiento Funcional 05.

<b>Identificación del requerimiento:</b>	<b>RF05</b>
<b>Nombre del Requerimiento:</b>	Integración Interna (II)
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	Se permite al DBA consultar información ingresada en el sistema por los Operadores mediante los Formularios, durante la ejecución del proceso.
<b>Descripción del requerimiento:</b>	<p>Si el usuario Administrador, durante el diseño o rediseño de un proceso, decide que se necesita consultar información que se va a ingresar en la BD del sistema, enviará la tarea al DBA, indicando los datos que se deben consultar.</p> <p>El DBA recibe el mensaje y realiza la consulta de los datos requeridos, en el rango o plazo de fechas necesarias. Podrá consultar datos específicos, en caso de que requieran filtro o parámetros de entrada.</p> <p>Cuando la II esté lista, se notifica al administrador el cumplimiento de la actividad.</p> <p>En caso de que la II sea aprobada, será guardada en la BD para su posterior utilización, durante la ejecución del proceso.</p> <p>En caso de que la II sea rechazada, el Administrador podrá indicar el motivo por el cual tomó esa decisión y el DBA deberá seguir esas observaciones y volver a realizar la tarea.</p>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 27. Requerimiento Funcional 06.

<b>Identificación del requerimiento:</b>	<b>RF06</b>
<b>Nombre del Requerimiento:</b>	Integración Externa (IE)
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	Se permite al DBA consultar información brindada por servicios web externos al Workflow.
<b>Descripción del requerimiento:</b>	<p>Si el usuario Administrador, durante el diseño o rediseño de un proceso, decide que se necesita consultar información almacenada en BD externas al sistema y que puedan ser accedidas mediante URL, enviará la tarea al DBA.</p> <p>El DBA recibe el mensaje y realiza la consulta de los datos requeridos. Podrá consultar datos específicos, en caso de que requieran filtro o parámetros de entrada.</p> <p>Cuando la IE esté lista, se notifica al administrador el cumplimiento de la actividad.</p> <p>En caso de que la IE sea aprobada, será guardada en la BD para su posterior utilización, durante la ejecución del proceso.</p> <p>En caso de que la IE sea rechazada, el Administrador podrá indicar el motivo por el cual tomó esa decisión y el DBA deberá seguir esas observaciones y volver a realizar la tarea.</p>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 28. Requerimiento Funcional 07.

<b>Identificación del requerimiento:</b>	<b>RF07</b>
<b>Nombre del Requerimiento:</b>	Vista de datos
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	Se permite al usuario Diseñador generar vistas a partir de los datos obtenidos en una integración, ya sea interna o externa.
<b>Descripción del requerimiento:</b>	<p>Si el usuario Administrador, durante el diseño o rediseño de un proceso, decide que se necesita presentar información consultada mediante integración interna o externa y que además tenga una interfaz personalizada, enviará la tarea al Diseñador, indicando lo solicitado.</p> <p>El Diseñador recibe el mensaje y realiza la respectiva Vista de Datos.</p> <p>Cuando la Vista de Datos esté lista, se notifica al administrador el cumplimiento de la actividad.</p> <p>En caso de que la Vista de Datos sea aprobada, será guardada en la BD para su posterior utilización, durante la ejecución del proceso.</p> <p>En caso de que la Vista de Datos sea rechazada, el Administrador podrá indicar el motivo por el cual tomó esa decisión y el Diseñador deberá seguir esas observaciones y volver a realizar la tarea.</p>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 29.** Requerimiento Funcional 08.

<b>Identificación del requerimiento:</b>	<b>RF08</b>
<b>Nombre del Requerimiento:</b>	Reporte de formularios
<b>Requerimiento Dependiente:</b>	RF01
<b>Características:</b>	Se permite al usuario Diseñador generar reportes en formato PDF a partir de los datos obtenidos en una integración, ya sea interna o externa.
<b>Descripción del requerimiento:</b>	<p>Si el usuario Administrador, durante el diseño o rediseño de un proceso, decide que se necesita presentar información consultada mediante integración interna o externa y que además se genere automáticamente en formato PDF, enviará la tarea al Diseñador, indicando lo solicitado.</p> <p>El Diseñador recibe el mensaje y realiza la respectiva Vista de Datos.</p> <p>Cuando el Reporte de Formulario esté listo, se notifica al administrador el cumplimiento de la actividad.</p> <p>En caso de que el Reporte de Formulario sea aprobado, será guardada en la BD para su posterior utilización, durante la ejecución del proceso.</p> <p>En caso de que el Reporte de Formulario sea rechazado, el Administrador podrá indicar el motivo por el cual tomó esa decisión y el Diseñador deberá seguir esas observaciones y volver a realizar la tarea.</p>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 30.** Requerimiento Funcional 09.

<b>Identificación del requerimiento:</b>	<b>RF09</b>
<b>Nombre del Requerimiento:</b>	Aprobar o rechazar tareas durante el diseño del proceso.
<b>Requerimiento Dependiente:</b>	RF02, RF03, RF04, RF05, RF06, RF07
<b>Características:</b>	El usuario Administrador podrá aprobar o rechazar las tareas enviadas por los demás usuarios (Diseñador, Modelador, DBA).
<b>Descripción del requerimiento:</b>	<p>El Administrador recibe alguna tarea finalizada por otro usuario (Diseñador, Modelador, DBA), con la facilidad de probar o ejecutar la tarea, por ejemplo: en el caso de una IE puede verificar que la consulta al servicio web externo devuelva los datos requeridos.</p> <p>Si el Administrador decide que la tarea es correcta, procede a su respectiva aprobación.</p> <p>Si el Administrador decide que la tarea es incorrecta, procede a rechazarla y enviarle al usuario responsable el motivo de su decisión, para que pueda volver a realizarla.</p>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 31. Requerimiento Funcional 10.

<b>Identificación del requerimiento:</b>	<b>RF10</b>
<b>Nombre del Requerimiento:</b>	Definición de un proceso.
<b>Requerimiento Dependiente:</b>	RF01, RF03, RF09
<b>Características:</b>	Registro de un nuevo proceso en la BD.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Luego de que las tareas realizadas por el DBA, Diseñador y Modelador durante el diseño del proceso han finalizado con éxito, el core se encargará de analizar y capturar desde el diagrama XML que haya guardado previamente el Modelador, los elementos que intervengan en el proceso: tareas, subprocesos, entre otros y los almacenará en las respectivas tablas de la base de datos.</li> <li>✓ Una vez definido, procesado y almacenado, el nuevo proceso estará listo la respectiva asignación de recursos.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 32. Requerimiento Funcional 11.

<b>Identificación del requerimiento:</b>	<b>RF11</b>
<b>Nombre del Requerimiento:</b>	Asignación de recursos al diagrama.
<b>Requerimiento Dependiente:</b>	RF10
<b>Características:</b>	Se asignarán los recursos (responsables, formularios, vista de datos, plazo estimado, etc.) a las actividades, compuertas y eventos del diagrama realizado y aprobado previamente.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ El usuario administrador hará la asignación de los respectivos recursos y se almacenarán en la BD, para la ejecución de los elementos del diagrama.</li> <li>✓ Los recursos que pueden ser asignados en las tarea son: elementos como formularios, reportes y vistas de datos de formularios o de integraciones (internas o externas); usuarios responsables (cargos o usuario específico, incluido el usuario que ha instanciado el proceso); y plazo o tiempo (en días) estimado para ejecutar la tarea.</li> <li>✓ El recurso que puede ser asignado en los eventos de inicio es el usuario responsable para instanciar el proceso, pudiendo ser un usuario específico o un cargo administrativo.</li> <li>✓ Los eventos de fin no requieren recursos, ya que se ejecutan automáticamente.</li> <li>✓ Los recursos que pueden ser asignados a las compuertas son: reglas de condición o usuario encargado de tomar una decisión.</li> <li>✓ Durante la asignación, en caso de que el Administrador ingrese información errónea a algún elemento del diagrama, podrá realizar la actualización en tiempo real.</li> <li>✓ Mientras se va realizando la asignación de los recursos, automáticamente se va actualizando el progreso alcanzado, hasta llegar a un 100%.</li> <li>✓ Una vez que se ha finalizado la asignación, el proceso quedará automáticamente listo para ser instanciado.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 33. Requerimiento Funcional 12.

<b>Identificación del requerimiento:</b>	<b>RF12</b>
<b>Nombre del Requerimiento:</b>	Instancia de un proceso.
<b>Requerimiento Dependiente:</b>	RF11
<b>Características:</b>	Los usuarios operadores podrán ejecutar (instanciar) los procesos a los que hayan sido asignados.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Una vez que el proceso ha sido configurado, el mismo podrá ser utilizado (instanciado) por los usuarios correspondientes cuando sea necesario.</li> <li>✓ Cuando un proceso es instanciado se almacena un registro en la BD, indicando que usuario lo ha inicializado, registrando además la fecha de la instancia.</li> <li>✓ El core se encargará de obtener la ruta o el orden de ejecución de las tareas de acuerdo con el flujo del proceso previamente establecido, con sus respectivos responsables, fechas de entrega (en caso de tener), entre otros elementos.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 34. Requerimiento Funcional 13.

<b>Identificación del requerimiento:</b>	<b>RF13</b>
<b>Nombre del Requerimiento:</b>	Eliminar instancia de un proceso.
<b>Requerimiento Dependiente:</b>	RF12, RF24
<b>Características:</b>	Los usuarios Supervisores podrán eliminar la instancia de un proceso, cuando haya ocurrido un error fatal.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ En caso de que el core identifique algún error fatal durante la ejecución del proceso, automáticamente dará al Supervisor la opción de eliminar dicha instancia.</li> <li>✓ Cuando el core reciba la confirmación de la orden por parte del Usuario de eliminar la instancia, eliminará por completo todos los registros relacionados a la misma: tareas ejecutadas, información llenada a través de los formularios, documentos recolectados, etc.</li> <li>✓ La eliminación de una instancia no se realiza por borrado lógico, es decir que es una acción que no puede deshacerse.</li> </ul>
<b>Prioridad del requerimiento:</b>	Media
<b>Elaboración:</b>	Autores

Cuadro 35. Requerimiento Funcional 14.

<b>Identificación del requerimiento:</b>	<b>RF14</b>
<b>Nombre del Requerimiento:</b>	Reiniciar instancia de un proceso.
<b>Requerimiento Dependiente:</b>	RF12
<b>Características:</b>	Los usuarios Supervisores podrán reiniciar la instancia de un proceso, cuando haya ocurrido un error fatal.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ En caso de que el core identifique algún error fatal durante la ejecución del proceso, automáticamente dará al Supervisor la opción de reiniciar dicha instancia.</li> <li>✓ Cuando el core reciba la confirmación de la orden por parte del Usuario de reiniciar la instancia, eliminará por completo todos los registros relacionados a la misma: tareas ejecutadas, información llenada a través de los formularios, documentos recolectados, etc. Y procederá a ejecutar la instancia desde el inicio.</li> <li>✓ La eliminación de una instancia realizada automáticamente cuando ésta es reiniciada, no se realiza por borrado lógico, es decir que es una acción que no puede deshacerse.</li> </ul>
<b>Prioridad del requerimiento:</b>	Media
<b>Elaboración:</b>	Autores

Cuadro 36. Requerimiento Funcional 15.

<b>Identificación del requerimiento:</b>	<b>RF15</b>
<b>Nombre del Requerimiento:</b>	Historial de instancia de un proceso.
<b>Requerimiento Dependiente:</b>	RF12, RF13, RF14
<b>Características:</b>	El core llevará el registro de todos los cambios de estado de una instancia.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Cada vez que el estado de una instancia sufra cambios, se almacenará el respectivo registro en la BD, como mecanismo de control.</li> <li>✓ Se entiende por cambio lo siguiente: cuando se ejecuta o instancia un proceso, el estado es "en ejecución"; cuando se detiene la instancia porque hay una función de llamada a la instancia de otro proceso, el estado es "en pausa", cuando ha sido eliminada del sistema, el estado es "eliminado", cuando la instancia finaliza exitosamente, el estado es "finalizado" y si termina por algún error el estado es "terminado".</li> <li>✓ Además, deberá registrarse la fecha exacta del cambio producido y el trámite o actividad donde se presentó el cambio de estado, en caso de ser necesario.</li> </ul>
<b>Prioridad del requerimiento:</b>	Media
<b>Elaboración:</b>	Autores

Cuadro 37. Requerimiento Funcional 16.

<b>Identificación del requerimiento:</b>	RF16
<b>Nombre del Requerimiento:</b>	Rediseño de un proceso.
<b>Requerimiento Dependiente:</b>	RF10, RF11
<b>Características:</b>	Los Administradores podrán hacer cambios al diseño o flujo de un proceso ya creado, asignar nuevos atributos, etc.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Los procesos almacenados en la BD tiene la opción de ser modificados o rediseñados (versionamiento).</li> <li>✓ Un proceso puede ser rediseñado solamente si no hay instancias del mismo en ejecución.</li> <li>✓ Una vez rediseñado el proceso el usuario podrá guardar la nueva versión, se debe tener en cuenta que la versión anterior pasará a estar en modo inactivo.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

### 1.5.2 ACTIVIDADES

Las actividades representan la lógica para una tarea específica en un proceso de flujo de trabajo. Se dispondrán de las funciones de inicio y cancelación de procesos y realización de actividades, mediante servicios web, como mecanismo predilecto de comunicación entre procesos.

Una actividad estará representada en un flujo de trabajo como un nodo. El core dispondrá de los siguientes tipos de nodos, con su respectiva descripción:

- ✓ **Atributos comunes para las actividades de flujo de trabajo:** Cada actividad de flujo de trabajo tendrá un conjunto de atributos que se deben configurar para que el flujo de trabajo funcione adecuadamente.
- ✓ **Nodos de inicio y de fin:** El nodo de inicio define el comienzo de un flujo de trabajo y el nodo de fin define el final de un flujo de trabajo.
- ✓ **Nodo de operación:** Invocará una operación existente desde un flujo de trabajo.
- ✓ **Nodo de tarea:** Permitirá utilizar una tarea, la cual representa una sola unidad de trabajo que no es o no se puede dividir a un mayor nivel de detalle de procesos de negocio.

- ✓ **Nodo de subproceso:** Ejecutará un flujo de trabajo de titularidad, desde otro.

A continuación se presentan los RF relacionados con las actividades (Cuadros 29 - 31).

**Cuadro 38.** Requerimiento Funcional 17.

<b>Identificación del requerimiento:</b>	<b>RF17</b>
<b>Nombre del Requerimiento:</b>	Supervisión de tareas (trámite) del proceso.
<b>Requerimiento Dependiente:</b>	RF12
<b>Características:</b>	El usuario Supervisor podrá inspeccionar el estado de las tareas de los diversos procesos que estén a su cargo.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Las diferentes tareas que intervienen en el proceso instanciado por un usuario, deberán ser atendidas, a tiempo, por los respectivos responsables.</li> <li>✓ El Usuario Supervisor podrá verificar el cumplimiento de todas las tareas del proceso al que haya sido asignado.</li> <li>✓ Si una tarea no es completada a tiempo, se notificará al Supervisor sobre lo sucedido, informándose el título y responsable de la misma.</li> <li>✓ El Supervisor podrá modificar el plazo de entrega de las tareas.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 39.** Requerimiento Funcional 18.

<b>Identificación del requerimiento:</b>	<b>RF18</b>
<b>Nombre del Requerimiento:</b>	Ejecución de las actividades.
<b>Requerimiento Dependiente:</b>	RF12
<b>Características:</b>	Las ejecuciones de las actividades serán realizadas de acuerdo al flujo establecido en el diseño del proceso instanciado.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Luego de que el proceso es instanciado, el núcleo tomará como entrada un flujo de proceso y mantendrá el estado de la ejecución de las actividades.</li> <li>✓ El flujo de una actividad a otra estará controlado por la lógica condicional (codificación) en las transiciones y la información de configuración de la actividad.</li> <li>✓ Para dar cumplimiento a cada actividad que interviene en el flujo de trabajo, el core delegará y distribuirá las actividades a realizar entre los actores humanos y aplicaciones software respectivas.</li> <li>✓ Se podrán definir flujos de trabajo seriales y paralelos.</li> <li>✓ La asignación al responsable para la ejecución de cada actividad se realizará de forma automática, según como se haya indicado en el diagrama de procesos, pudiendo ser: cuando se cumple la actividad predecesora o al instanciarse el proceso.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 40.** Requerimiento Funcional 19.

<b>Identificación del requerimiento:</b>	<b>RF19</b>
<b>Nombre del Requerimiento:</b>	Reiniciar actividad.
<b>Requerimiento Dependiente:</b>	RF12, RF24
<b>Características:</b>	El Supervisor podrá reiniciar una actividad, en caso de que se haya excedido el tiempo límite y la actividad no haya sido finalizada.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ En caso de que una actividad no haya sido finalizada en el tiempo indicado, el core dará la opción al Supervisor de que se reinicie esa actividad.</li> <li>✓ Una vez que el Supervisor emita la orden, el core se encargará de recalcular una nueva fecha límite a esa tarea, según el límite de días establecido durante la asignación y notificará al Operador para que proceda con la respectiva ejecución.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b> Autores	

### 1.5.3 PARTICIPANTES DE LA ACTIVIDAD

Los participantes serán los usuarios que se hayan asignado para interactuar con las actividades de un proceso de flujo de trabajo, como aprobaciones, solicitudes de información, etc. A continuación se detallan los RF referentes a los participantes de un proceso (Cuadros 32 - 34).

**Cuadro 41.** Requerimiento Funcional 20.

<b>Identificación del requerimiento:</b>	<b>RF20</b>
<b>Nombre del Requerimiento:</b>	Gestión de cargos administrativos.
<b>Requerimiento Dependiente:</b>	Ninguno.
<b>Características:</b>	Crear, modificar y eliminar los cargos administrativos.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Se podrán crear los diferentes cargos del personal administrativo de la ESPAM MFL, para que posteriormente el Administrador del proceso pueda asignar los respectivos responsables a cada cargo.</li> <li>✓ Cuando existan cambios en el área administrativa, se podrá actualizar o eliminar los cargos existentes.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b> Autores	

**Cuadro 42.** Requerimiento Funcional 21.

<b>Identificación del requerimiento:</b>	RF21
<b>Nombre del Requerimiento:</b>	Definición de participantes.
<b>Requerimiento Dependiente:</b>	RF11, RF20.
<b>Características:</b>	Determinar los responsables de las diferentes acciones en una actividad.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ En caso de que durante la asignación de recursos, se determine el responsable por cargo administrativo, automáticamente el core asignará a diversos usuarios, de acuerdo a su relación con el proceso o workflow en ejecución.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

**Cuadro 43.** Requerimiento Funcional 22.

<b>Identificación del requerimiento:</b>	RF22
<b>Nombre del Requerimiento:</b>	Reasignación de participantes
<b>Requerimiento Dependiente:</b>	RF21
<b>Características:</b>	Garantizar el poder hacer cambios de responsables de las tareas o trámites del proceso en caso de ser necesario.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Durante la ejecución de un proceso, si el Supervisor observa que existen cambios importantes en el mismo, que afectan directamente a los responsables (Operadores) de las actividades, podrá reasignarlos, cuando lo crea conveniente.</li> <li>✓ Durante la creación o diseño de un proceso, si el Administrador requiere cambiar el responsable de una tarea, por motivos internos de la organización, podrá reemplazar a dicho usuario (DBA, Diseñador, Modelador) por otro.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

#### 1.5.4 CONTROL

El núcleo inspeccionará la evolución de cada una de las actividades de un proceso, en caso de que no se completen a tiempo o que existan contradicciones entre la información solicitada y la información brindada, por parte de los usuarios y se presentarán notificaciones a los responsables de cada tarea y al Supervisor del proceso, generando de forma automática los respectivos errores durante la ejecución y emitiendo las opciones necesarias para efectuar las posibles soluciones.

A continuación se presentan los RF involucrados con el control que se tendrá en el núcleo del Workflow (Cuadros 35 - 37).

Cuadro 44. Requerimiento Funcional 23.

<b>Identificación del requerimiento:</b>	<b>RF23</b>
<b>Nombre del Requerimiento:</b>	Mecanismos de Control
<b>Requerimiento Dependiente:</b>	RF18
<b>Características:</b>	Establecer mecanismos que permitan controlar la ejecución de una actividad.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Cuando un usuario diseñe, o mejore un proceso, puede especificar límites en ciertas tareas para la aprobación o realización de la misma, la cual será monitoreada constantemente con el fin de cumplir con dicha especificación.</li> <li>✓ El mecanismo de control implementado en el workflow se realizará mediante una tarea programada (Sql Server Agent) que se encargará de monitorear el cumplimiento de los tiempos establecidos y emitirá una alerta o notificación del estado de la tarea al usuario supervisor del proceso y al responsable de la misma.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 45. Requerimiento Funcional 24.

<b>Identificación del requerimiento:</b>	<b>RF24</b>
<b>Nombre del Requerimiento:</b>	Errores durante la ejecución de un proceso.
<b>Requerimiento Dependiente:</b>	RF18
<b>Características:</b>	Presentar al Administrador los posibles errores que se puedan dar durante la ejecución de un proceso y las alternativas de solución.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Durante la ejecución de un proceso se pueden presentar 2 tipos de errores, detallados a continuación.</li> <li>✓ En caso de una compuerta que toma decisiones en base a reglas no encuentra un camino que cumpla con las condiciones especificadas y no tiene un "default" o camino por defecto, generará un "error fatal" permitiendo al Supervisor reiniciar o eliminar la instancia.</li> <li>✓ En caso de que una tarea no se cumpla a tiempo, según el mecanismo de control establecido, el core generará un error por incumplimiento, permitiendo que el Supervisor reinicie la instancia.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

Cuadro 46. Requerimiento Funcional 25.

<b>Identificación del requerimiento:</b>	<b>RF25</b>
<b>Nombre del Requerimiento:</b>	Notificaciones
<b>Requerimiento Dependiente:</b>	RF18, RF23, RF24
<b>Características:</b>	Informar de las tareas y actividades existentes en el sistema, a los respectivos usuarios.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ Para la creación de nuevo proceso, existirán tareas que deberán ser notificadas en los plazos correspondientes a los responsables: Modelador, Diseñador de</li> </ul>

---

	formularios y DBA.
✓	Luego de que el proceso es aprobado e instanciado (ejecutado) por el Administrador se obtienen las rutas u orden de ejecución de las tareas, mismas que tienen un responsable (Operador) de ejecutarlas.
✓	El core debe ser capaz de notificar a dichos responsables mediante alertas o mensajes que hay una tarea que requiere su atención.
✓	Además, en caso de que el tiempo de ejecución de tareas haya excedido el plazo de entrega, deberá notificarse al Supervisor responsable del proceso, o en su defecto de las tareas previas a la creación del proceso.
✓	Cuando un proceso o las tareas previas a la creación del proceso han finalizado con éxito, se notifica al Administrador.
✓	Cuando la instancia de un proceso presente algún error, se notifica al Supervisor.

---

<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

---

### 1.5.5 REPORTE

Para facilitar la toma de decisiones a los Supervisores y Administradores de los diferentes procesos, el núcleo será capaz de procesar los datos necesarios que permitan la realización de reportes.

A continuación se detalla el RF que corresponde a la generación de reportes por parte del sistema (Cuadro 38).

**Cuadro 47.** Requerimiento Funcional 26.

---

<b>Identificación del requerimiento:</b>	RF26
<b>Nombre del Requerimiento:</b>	Reportes
<b>Requerimiento Dependiente:</b>	RF10, RF12
<b>Características:</b>	Garantizar que el core devuelva datos necesarios para analizar el funcionamiento del mismo.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ El core del workflow permitirá al usuario "Supervisor" y "Administrador" poder realizar consultas para obtener un reporte del estado de los procesos que ellos operen.</li> <li>✓ El core devolverá información del reporte para generar un documento en formato PDF.</li> </ul>

---

<b>Prioridad del requerimiento:</b>	Media
<b>Elaboración:</b>	Autores

---

### 1.5.6 INTEGRACIONES

Es necesario que el sistema pueda ser integrado a los demás sistemas de la ESPAM MFL, por lo cual se llevarán a cabo la integración de datos, a fin de que el workflow no quede aislado y, al contrario, pueda ser aprovechado al máximo por la organización.

A continuación se detalla el RF que permite que el núcleo pueda tener integración con otros sistemas (Cuadro 39).

**Cuadro 48.** Requerimiento Funcional 27.

<b>Identificación del requerimiento:</b>	<b>RF27</b>
<b>Nombre del Requerimiento:</b>	Integración de datos
<b>Requerimiento Dependiente:</b>	RF07, RF06, RF12
<b>Características:</b>	Permitir la integración de diferentes fuentes de datos.
<b>Descripción del requerimiento:</b>	<ul style="list-style-type: none"> <li>✓ El core podrá integrarse con las diferentes fuentes de datos de la ESPAM MFL (Integración Externa), a fin de que al realizar un ingreso de datos en una pantalla (o formulario), sea posible extraer, consultar y validar los datos contra dichas fuentes de datos.</li> <li>✓ La manera de acceder a datos externos es mediante los Web services.</li> <li>✓ Además podrá acceder mediante Integración Interna a la información propia del sistema, adquirida mediante los formularios del proceso.</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta
<b>Elaboración:</b>	Autores

### 1.6 REQUERIMIENTOS NO FUNCIONALES.

Hacen relación a las características del core del Workflow que aplican de manera general como un todo, más que a rasgos particulares del mismo. Estos requerimientos son adicionales a los requerimientos funcionales presentados en el tema anterior, y corresponden a aspectos tales como la disponibilidad, mantenibilidad, flexibilidad, seguridad, facilidad de uso, etc., los cuales se describen en esta sección.

## **1.6.1 ATRIBUTOS DE CALIDAD**

### **DESEMPEÑO**

Garantizará la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido, el core se encargará de gestionar la información almacenada en la base de datos del sistema, para que pueda ser consultada y actualizada permanente y simultáneamente, sin que se afecte el tiempo de respuesta.

El sistema estará en capacidad de dar respuesta al acceso de todos los usuarios y a los procesos con tiempo de respuesta aceptable y uniforme, en períodos de alta, media y baja demanda de uso del sistema.

### **DISPONIBILIDAD**

Mediante el uso de los verbos HTTP recomendados para la construcción de una API WEB REST, el sistema estará disponible, para que los clientes puedan consumirla, cuando lo requieran.

### **ESCALABILIDAD**

Estará en capacidad de permitir el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

### **FACILIDAD DE USO E INGRESO DE LA INFORMACIÓN**

Este aspecto no se contemplará en el desarrollo del core, ya que dependerá directamente del diseño de las interfaces.

### **FLEXIBILIDAD**

El core será flexible a los cambios que se presenten en los procesos administrativos de la ESPAM MFL, porque permitirá aplicar reingeniería a los procesos, durante la ejecución del mismo, facilitando el manejo de versiones de los mismos.

## **INSTALACIÓN**

Será fácil de instalar en los servidores de la ESPAM MFL, y en otros, cuando sea necesario.

## **MANTENIBILIDAD**

El core, permitirá que futuros desarrolladores puedan realizar cambios en el mismo, ya que la codificación estará documentada y disponible. Además, durante la codificación, se utilizarán estándares en la nomenclatura, facilitando el entendimiento y percepción del trabajo realizado.

## **SEGURIDAD**

El acceso al sistema estará restringido por el uso de claves asignadas a cada uno de los usuarios. Sólo podrán ingresar al sistema las personas que estén registradas, estos usuarios serán clasificados en varios tipos de usuarios (o roles) con acceso a las opciones de trabajo definidas para cada rol.

- ✓ El control de acceso permitirá asignar los perfiles para cada uno de los roles identificados.
- ✓ Respecto a la confidencialidad, el sistema estará en capacidad de rechazar accesos o modificaciones indebidos (no autorizados) a la información y proveer los servicios requeridos por los usuarios legítimos del sistema.
- ✓ El sistema contará con mecanismos que permitan el registro de actividades con identificación de los usuarios que los realizaron.
- ✓ El sistema contará con pistas de auditoría de las actividades que se realizan sobre el sistema con niveles razonables para su reconstrucción e identificación de los hechos.

## **1.7 REQUISITOS COMUNES DE LAS INTERFACES**

### **1.7.1 INTERFACES DE HARDWARE**

Será necesario disponer de un servidor en perfecto estado con las siguientes

características:

- ✓ Servidor Intel Xeon.
- ✓ 2.53 GHz
- ✓ RAM de 6 GB.
- ✓ 15 MB de cache, 64 bits
- ✓ Disco duro de 2 teras 7200 rpm SATA3
- ✓ Windows Server 2008 R2

### **1.7.2 INTERFACES DE SOFTWARE**

- ✓ Sistema Operativo: Windows server 2008.
- ✓ Base de Datos: Sql Server 2008 que incluya el servicio Sql Server Agent.
- ✓ IDE: Visual Studio 2015

### **1.7.3 INTERFACES DE COMUNICACIÓN**

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible. Por ejemplo, para transferir archivos o documentos deberán utilizarse protocolos existentes (FTP u otros convenientes).

## ANEXO 3 – CASOS DE USO



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

# UML

## Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL



**DIAGRAMAS UML:  
CASOS DE USO**

Dayana H. Bailón Delgado  
José S. Mera Cantos

2017

## FICHA DEL DOCUMENTO

El cuadro 1 se detalla las versiones que ha tenido este anexo indicando: fecha, versión, autores y por quien fue revisado.

**Cuadro 49.** Ficha del documento

<b>Fecha</b>	<b>Versión</b>	<b>Autores</b>	<b>Verificado por</b>
20/10/2016	1.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira

## CONTENIDO GENERAL

FICHA DEL DOCUMENTO .....	167
CONTENIDO GENERAL.....	168
CONTENIDO DE CUADROS Y FIGURAS.....	170
DOCUMENTACIÓN DE LOS CASOS DE USO .....	171
1.    DIAGRAMAS UML: CASOS DE USO .....	171
1.1.    DIAGRAMA 1: DEFINICIÓN DE TAREAS PARA LA CREACIÓN DE UN PROCESO .....	171
1.2.    DIAGRAMA 2: SUPERVISIÓN DE TAREAS PARA DISEÑO DEL PROCESO	172
1.3.    DIAGRAMA 3: DEFINICIÓN DEL FLUJO (DIAGRAMA) .....	173
1.4.    DIAGRAMA 4: DISEÑADOR DE FORMULARIOS .....	173
1.5.    DIAGRAMA 5: INTEGRACIÓN INTERNA (II).....	174
1.6.    DIAGRAMA 6: INTEGRACIÓN EXTERNA (IE) .....	175
1.7.    DIAGRAMA 7: VISTA DE DATOS .....	175
1.8.    DIAGRAMA 8: REPORTE DE FORMULARIOS .....	176
1.9.    DIAGRAMA 9: APROBAR O RECHAZAR TAREAS DURANTE EL DISEÑO DEL PROCESO.....	177
1.10.    DIAGRAMA 10: DEFINICIÓN DE UN PROCESO .....	177
1.11.    DIAGRAMA 11: ASIGNACIÓN DE RECURSOS AL DIAGRAMA...	178
1.12.    DIAGRAMA 12: INSTANCIA DE UN PROCESO.....	178
1.13.    DIAGRAMA 13: ELIMINAR INSTANCIA DE UN PROCESO.....	179
1.14.    DIAGRAMA 14: REINICIAR INSTANCIA DE UN PROCESO .....	179
1.15.    DIAGRAMA 15: HISTORIAL DE INSTANCIA DE UN PROCESO..	180
1.16.    DIAGRAMA 16: REDISEÑO DE UN PROCESO .....	180
1.17.    DIAGRAMA 17: SUPERVISIÓN DE TAREAS (TRÁMITE) DEL PROCESO	181
1.18.    DIAGRAMA 18: EJECUCIÓN DE LAS ACTIVIDADES .....	182
1.19.    DIAGRAMA 19: REINICIAR ACTIVIDAD .....	182
1.20.    DIAGRAMA 20: GESTIÓN DE CARGOS ADMINISTRATIVOS .....	183
1.21.    DIAGRAMA 21: DEFINICIÓN DE PARTICIPANTES.....	183

1.22.	DIAGRAMA 22: REASIGNACIÓN DE PARTICIPANTES.....	184
1.23.	DIAGRAMA 23: MECANISMOS DE CONTROL.....	184
1.24.	DIAGRAMA 24: ERRORES DURANTE LA EJECUCIÓN DE UN PROCESO	184
1.25.	DIAGRAMA 25: NOTIFICACIONES .....	185
1.26.	DIAGRAMA 26: REPORTES.....	185
1.27.	DIAGRAMA 27: INTEGRACIÓN DE DATOS .....	186

## CONTENIDO DE CUADROS Y FIGURAS

<b>Cuadro 1.</b> Ficha del documento.....	167
<b>Cuadro 2.-</b> Actores y participación.....	171
<b>Figura 1.</b> Diagrama UML del RF01.....	172
<b>Figura 2.</b> Diagrama UML del RF02.....	172
<b>Figura 3.</b> Diagrama UML del RF03.....	173
<b>Figura 4.</b> Diagrama UML del RF04.....	174
<b>Figura 5.</b> Diagrama UML del RF05.....	174
<b>Figura 6.</b> Diagrama UML del RF06.....	175
<b>Figura 7.</b> Diagrama UML del RF07.....	176
<b>Figura 8.</b> Diagrama UML del RF08.....	176
<b>Figura 9.</b> Diagrama UML del RF09.....	177
<b>Figura 10.</b> Diagrama UML del RF10.....	177
<b>Figura 11.</b> Diagrama UML del RF11.....	178
<b>Figura 12.</b> Diagrama UML del RF12.....	179
<b>Figura 13.</b> Diagrama UML del RF13.....	179
<b>Figura 14.</b> Diagrama UML del RF14.....	180
<b>Figura 15.</b> Diagrama UML del RF15.....	180
<b>Figura 16.</b> Diagrama UML del RF16.....	181
<b>Figura 17.</b> Diagrama UML del RF17.....	181
<b>Figura 18.</b> Diagrama UML del RF18.....	182
<b>Figura 19.</b> Diagrama UML del RF19.....	182
<b>Figura 20.</b> Diagrama UML del RF20.....	183
<b>Figura 21.</b> Diagrama UML del RF21.....	183
<b>Figura 22.</b> Diagrama UML del RF22.....	184
<b>Figura 23.</b> Diagrama UML del RF25.....	185
<b>Figura 24.</b> Diagrama UML del RF26.....	186
<b>Figura 25.</b> Diagrama UML del RF27.....	186

## DOCUMENTACIÓN DE LOS CASOS DE USO

Desde el punto de vista del cliente final, se detalla el comportamiento del núcleo del Workflow, se especifica mediante actores la participación e interacción de los diferentes roles de usuarios y se puntualiza el funcionamiento de los procesos del sistema, en la documentación de cada caso de uso.

**ACTORES:** Entidades externas al sistema que guardan una relación con éste y que le demandan una funcionalidad, pudiendo ser actores humanos o sistemas externos (Cuadro 3).

**Cuadro 50.** Actores y participación.

ACTOR	DESCRIPCIÓN
<b>Administrador</b>	Puede aprobar o no un nuevo proceso, ejecutar (instanciar) un proceso, revisar el cumplimiento de todos los trámites que intervienen en el proceso que administra, reasignar responsables y realizar reportes de dichos procesos.
<b>Supervisor</b>	Puede verificar el cumplimiento de todos los trámites del proceso, reasignar responsables en las actividades y realizar reportes de los procesos.
<b>Operador</b>	Realiza únicamente los trámites (tareas) a los que ha sido asignado.
<b>Modelador</b>	Se encarga de modelar el diagrama (flujo) del proceso.
<b>Diseñador</b>	Se encarga de realizar el diseño de los formularios a utilizar en las actividades, además de vista de datos y reportes de estos formularios.
<b>DBA (Administrador de Base de Datos)</b>	Es el encargado de realizar integraciones a la BD interna así como a externas mediante servicios web, además de hacer vista de datos y reportes de dichas integraciones.
<b>BD (Base de Datos)</b>	Es la base de datos interna del sistema.
<b>BD externa</b>	Es cualquier Base de Datos externa, que no sea propia del sistema.

Elaboración: Autores

## 1. DIAGRAMAS UML: CASOS DE USO

### 1.1. DIAGRAMA 1: DEFINICIÓN DE TAREAS PARA LA CREACIÓN DE UN PROCESO

Consiste en la creación de las tareas que serán utilizadas como recursos en la ejecución del proceso. Se recibe un objeto JSON desde la aplicación cliente con los datos del proceso.

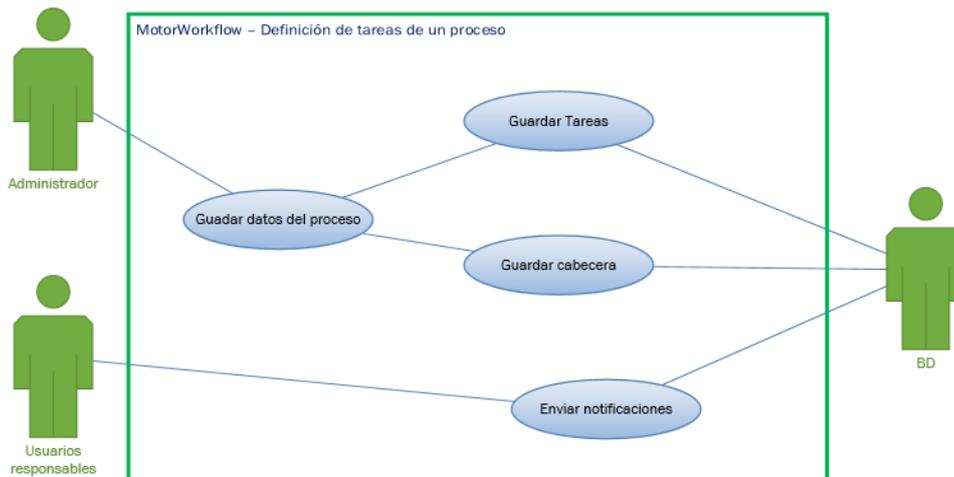


Figura 3. Diagrama UML del RF01.  
Elaboración: Autores

## 1.2. DIAGRAMA 2: SUPERVISIÓN DE TAREAS PARA DISEÑO DEL PROCESO

La supervisión de estas tareas es realizada por el usuario administrador, quien puede visualizar los datos del proceso y de las tareas, con el fin de ver cuales no han sido realizadas y su plazo de entrega está por terminar. Además, podrá realizar cambios de fecha límite del proceso o de cada una de las tareas, así como reasignar los usuarios responsables.

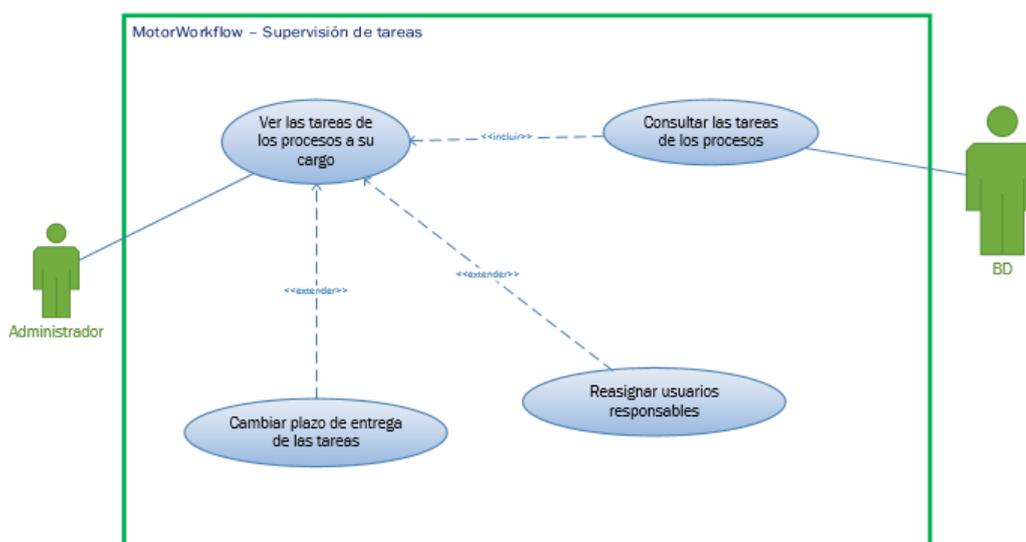


Figura 4. Diagrama UML del RF02.  
Elaboración: Autores

### 1.3. DIAGRAMA 3: DEFINICIÓN DEL FLUJO (DIAGRAMA)

Comprende en enviar el diagrama del flujo del proceso, el cual está a cargo del usuario modelador. El diagrama se valida y si es correcto se procede a hacer el envío al administrador.

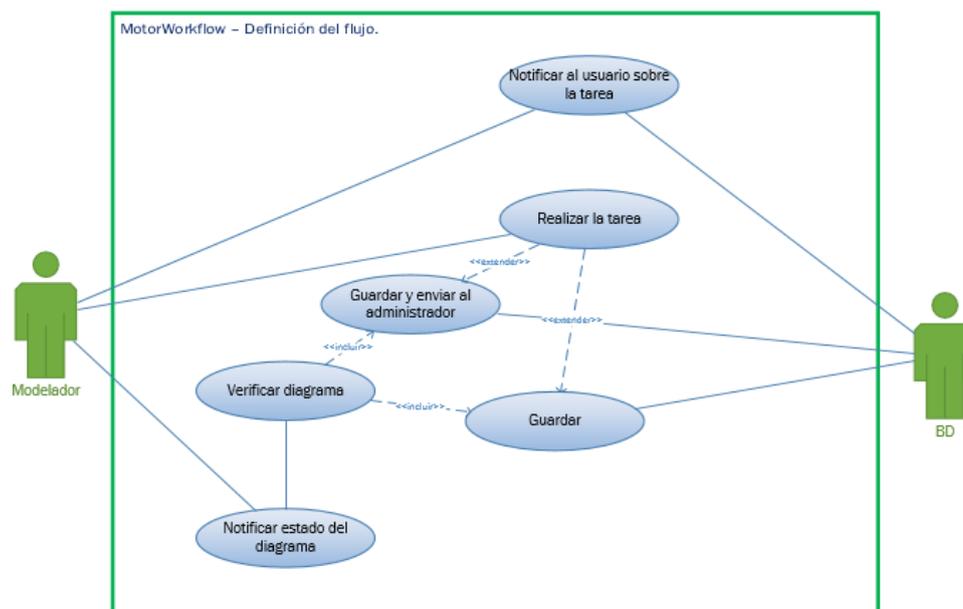


Figura 5. Diagrama UML del RF03  
Elaboración: Autores

### 1.4. DIAGRAMA 4: DISEÑADOR DE FORMULARIOS

Consiste en realizar un formulario por parte del usuario diseñador, el cual es verificado, a este diagrama se le pueden incluir máscaras o datos de integraciones. Una vez realizado el formulario se envía al administrador para su aprobación.

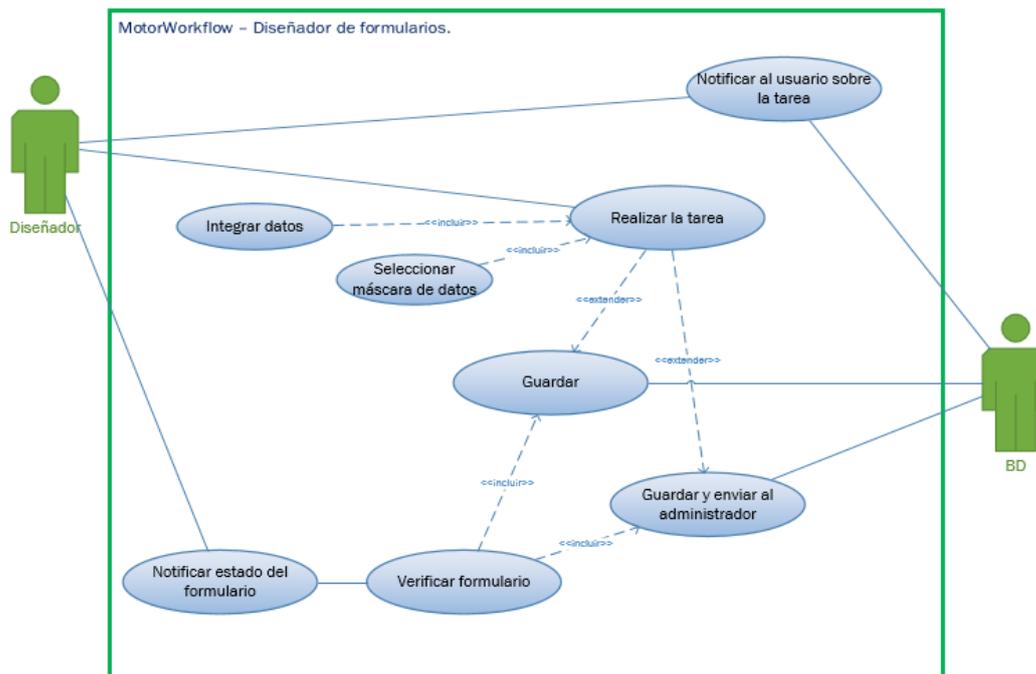


Figura 6. Diagrama UML del RF04  
Elaboración: Autores

### 1.5. DIAGRAMA 5: INTEGRACIÓN INTERNA (II)

En la Figura 7 se muestra el caso de uso en el cual un usuario DBA recibe la notificación de una tarea, la realiza probando la correcta ejecución de la integración, devolviendo los datos que van a ser integrados desde la BD.

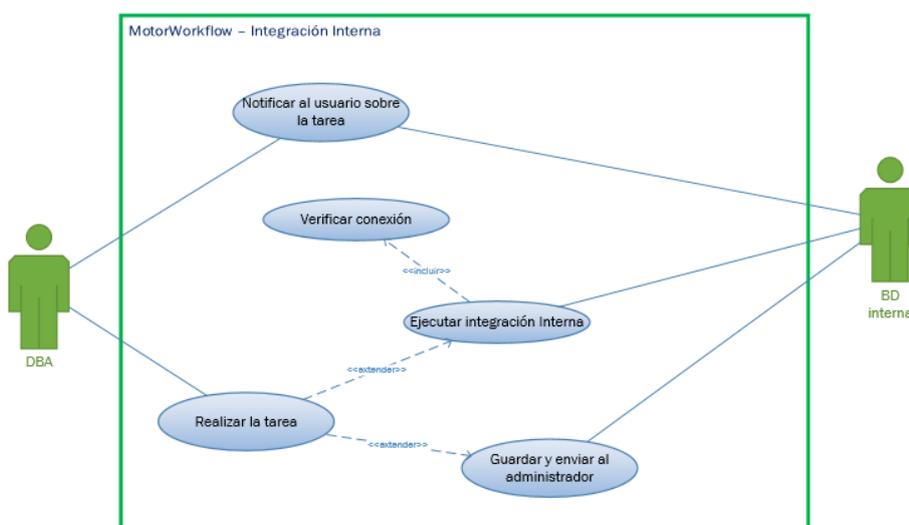


Figura 7. Diagrama UML del RF05.  
Elaboración: Autores

## 1.6. DIAGRAMA 6: INTEGRACIÓN EXTERNA (IE)

Como se muestra en la Figura 8 se notifica al usuario DBA sobre la tarea de integración externa que debe realizar, para lo cual cuando realiza la tarea esta ejecuta una prueba, consumiendo los datos de la BD externa.

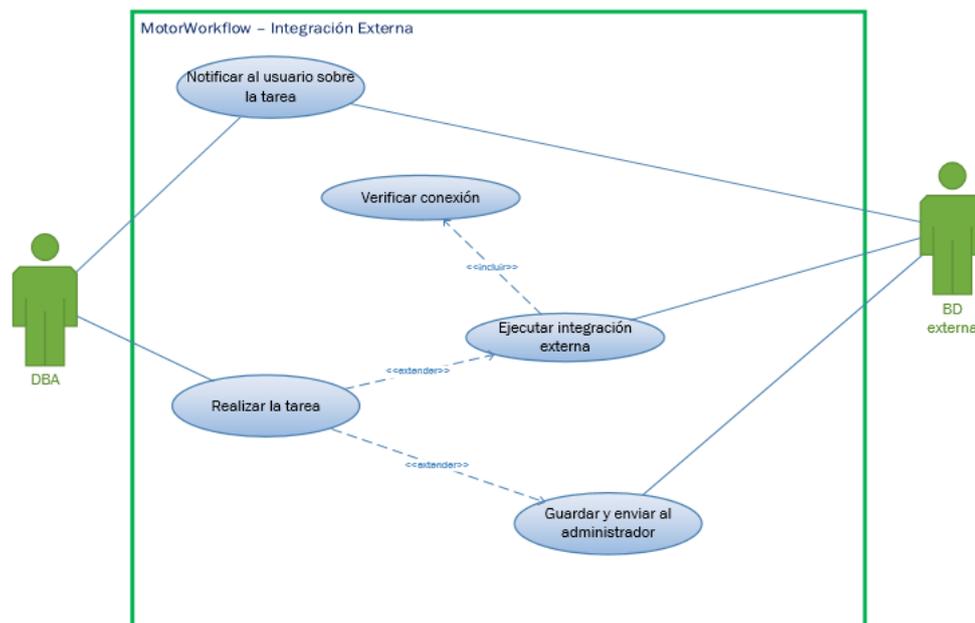


Figura 8. Diagrama UML del RF06  
Elaboración: Autores

## 1.7. DIAGRAMA 7: VISTA DE DATOS

La ejecución de este tipo de tarea inicia con la notificación al usuario DBA, el cual realiza la misma con las integraciones internas y externas aprobadas por el administrador, también existen vistas de datos para los formularios, esto lo realiza el usuario diseñador.

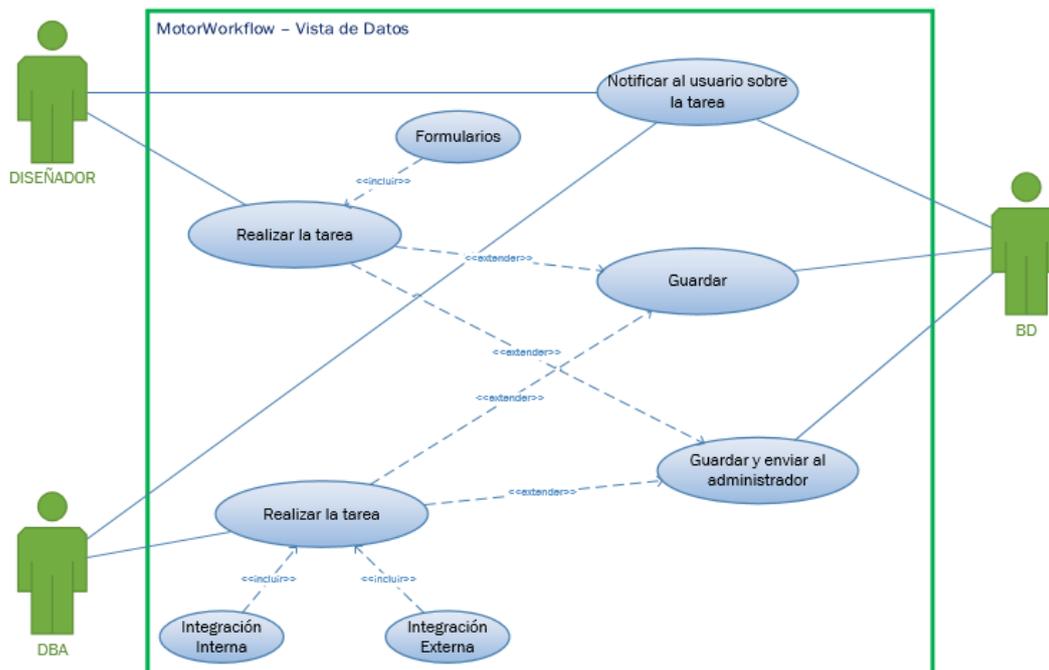


Figura 9. Diagrama UML del RF07  
Elaboración: Autores

### 1.8. DIAGRAMA 8: REPORTE DE FORMULARIOS

Inicia con la notificación al usuario DBA, el cual realiza el reporte de las integraciones internas y externas aprobadas por el administrador, además el usuario diseñador se encarga de realizar los reportes para los formularios, aprobados.

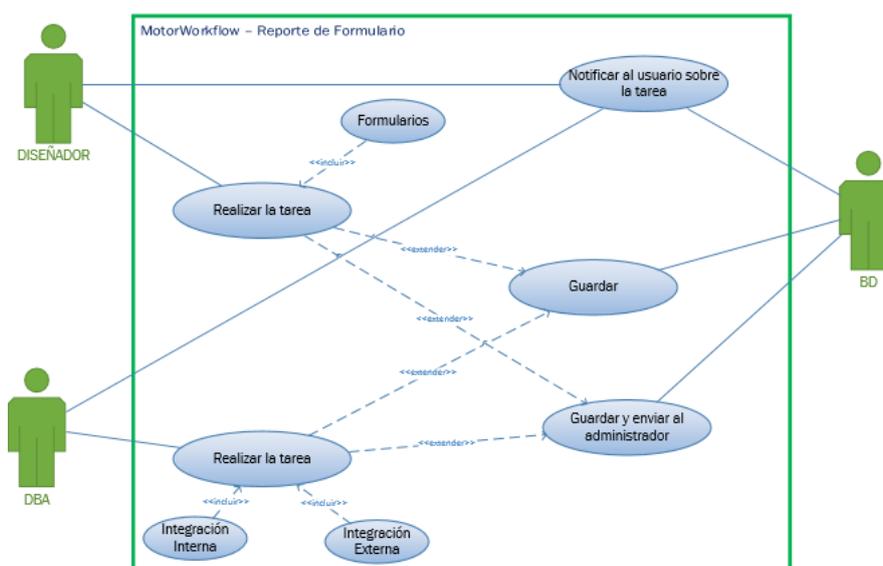


Figura 10. Diagrama UML del RF08  
Elaboración: Autores

### 1.9. DIAGRAMA 9: APROBAR O RECHAZAR TAREAS DURANTE EL DISEÑO DEL PROCESO

Cuando los usuarios encargados de realizar una tarea para la definición del proceso, se notifica al administrador del mismo para realizar la revisión de las mismas, el administrador se encarga de revisar la tarea, la cual puede ser aprobada o rechazada, si se rechaza se debe especificar el motivo.

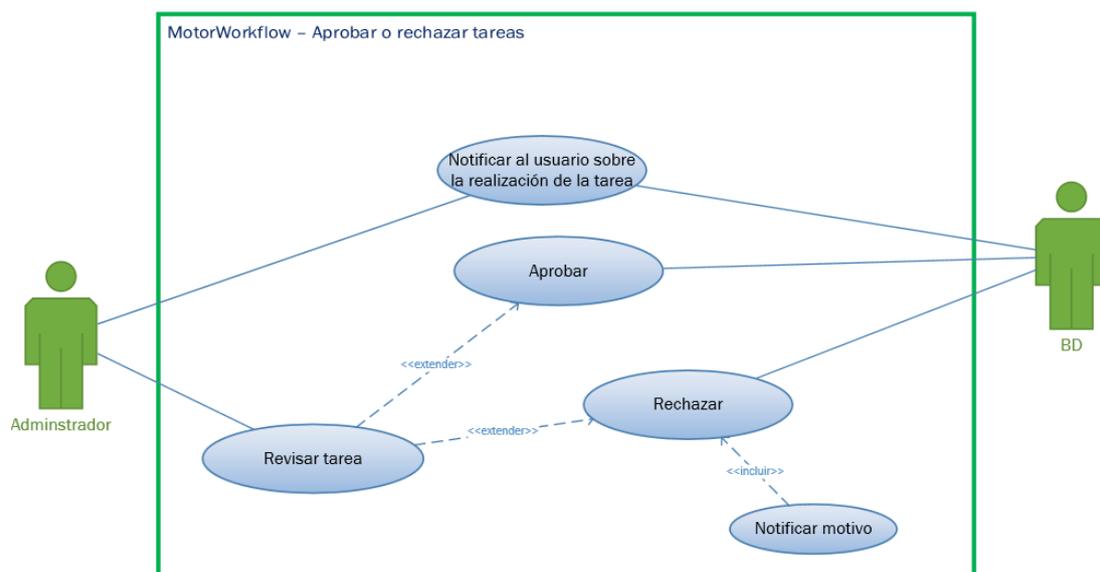


Figura 11. Diagrama UML del RF09  
Elaboración: Autores

### 1.10. DIAGRAMA 10: DEFINICIÓN DE UN PROCESO

El núcleo del workflow se encarga de manera interna de realizar la definición del proceso, esta consiste en analizar el XML del diagrama y obtener los elementos que lo conforman, para posteriormente guardarlos en la BD.

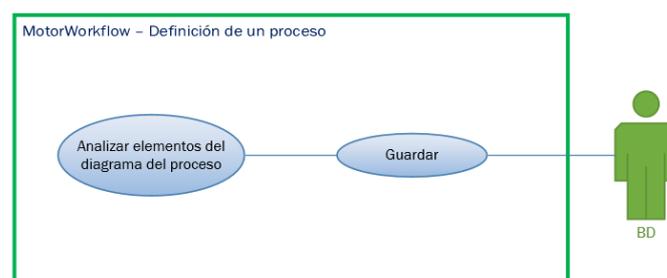
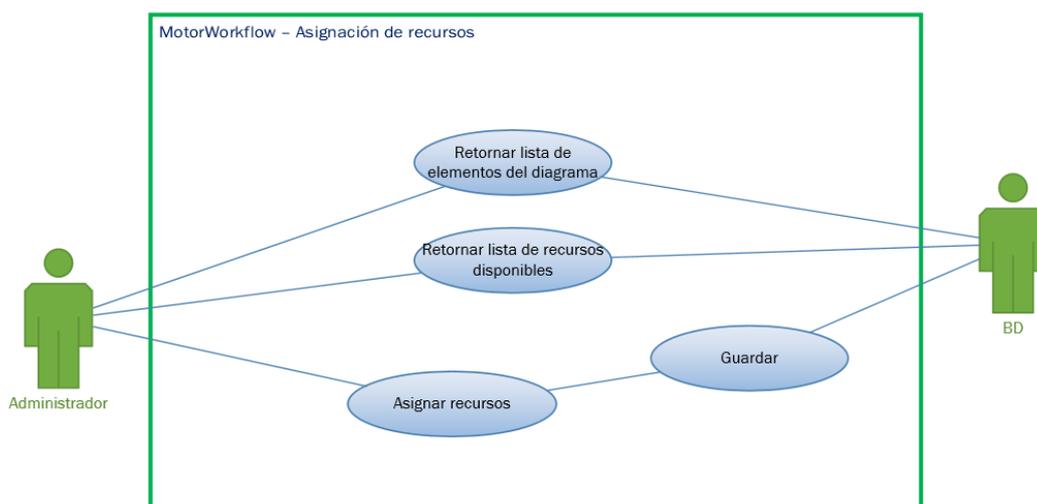


Figura 12. Diagrama UML del RF10  
Elaboración: Autores

### 1.11. DIAGRAMA 11: ASIGNACIÓN DE RECURSOS AL DIAGRAMA

La asignación de recursos está a cargo del administrador del proceso, el núcleo retorna a la aplicación cliente un listado de los elementos del diagrama, además de una lista con los recursos que se encuentran disponibles para el proceso.



**Figura 13.** Diagrama UML del RF11  
Elaboración: Autores

### 1.12. DIAGRAMA 12: INSTANCIA DE UN PROCESO

La instancia de los proceso estará a cargo de los operadores quienes ejecutan el (los) proceso(s), de manera interna se crea una nueva instancia con los registros de los trámites y las notificaciones a los usuarios correspondientes, según el flujo diseñado.

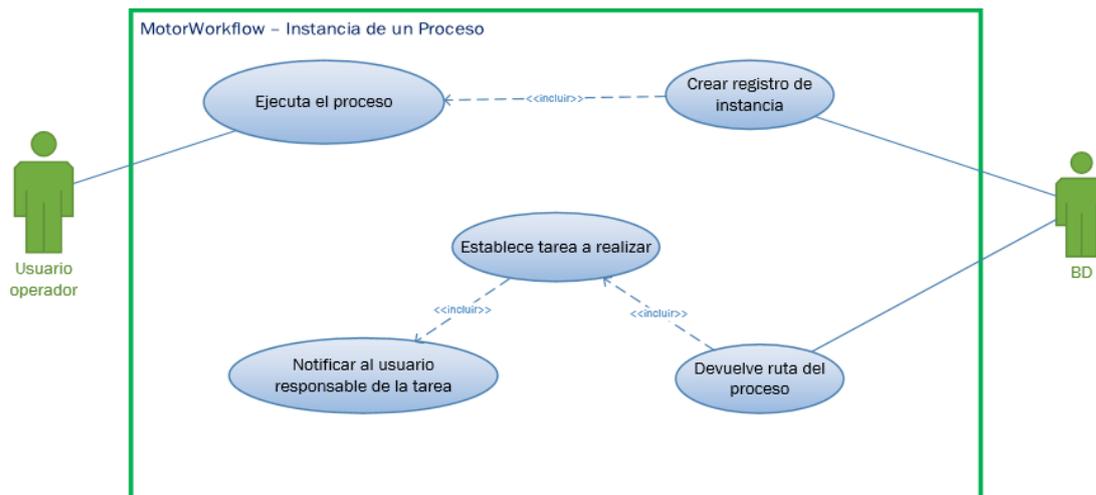


Figura 14. Diagrama UML del RF12  
Elaboración: Autores

### 1.13. DIAGRAMA 13: ELIMINAR INSTANCIA DE UN PROCESO

Consiste en permitir que el supervisor de un proceso tenga la opción de eliminar la instancia de un proceso, según su criterio.

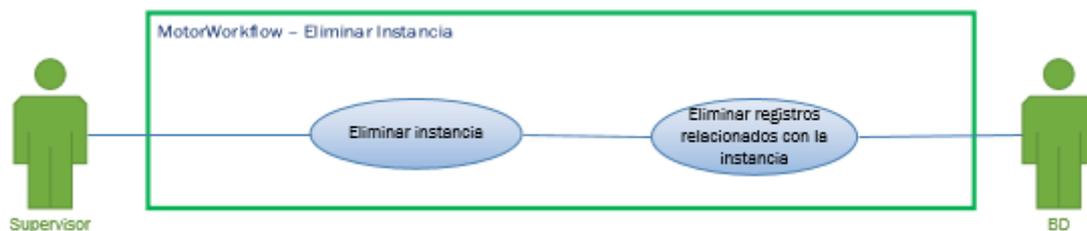
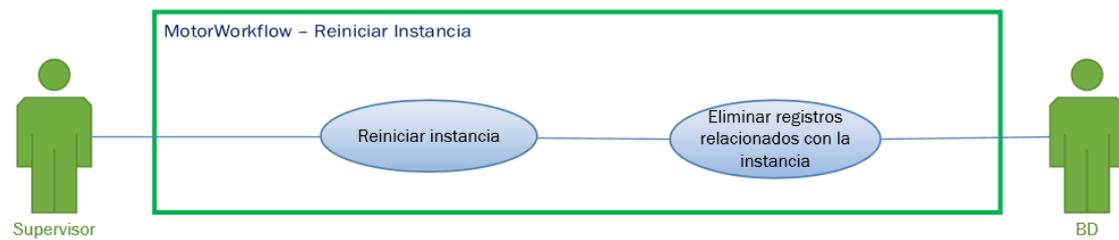


Figura 15. Diagrama UML del RF13  
Elaboración: Autores

### 1.14. DIAGRAMA 14: REINICIAR INSTANCIA DE UN PROCESO

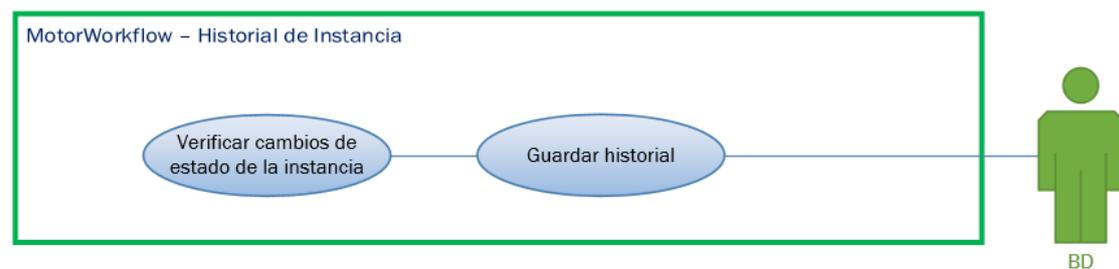
Permite al supervisor de un proceso reiniciar una instancia, esto sí ha ocurrido un error durante la ejecución del mismo.



**Figura 16.** Diagrama UML del RF14  
Elaboración: Autores

### 1.15. DIAGRAMA 15: HISTORIAL DE INSTANCIA DE UN PROCESO

El historial de una instancia se crea cuando esta cambia de estado, registrando en la BD los detalles del cambio.



**Figura 17.** Diagrama UML del RF15  
Elaboración: Autores

### 1.16. DIAGRAMA 16: REDISEÑO DE UN PROCESO

Para realizar el rediseño de un proceso, este no debe tener instancias en ejecución. Una vez que todas las instancias han finalizado se procede con el rediseño, que está a cargo del administrador, quien delegara las nuevas tareas a realizar.

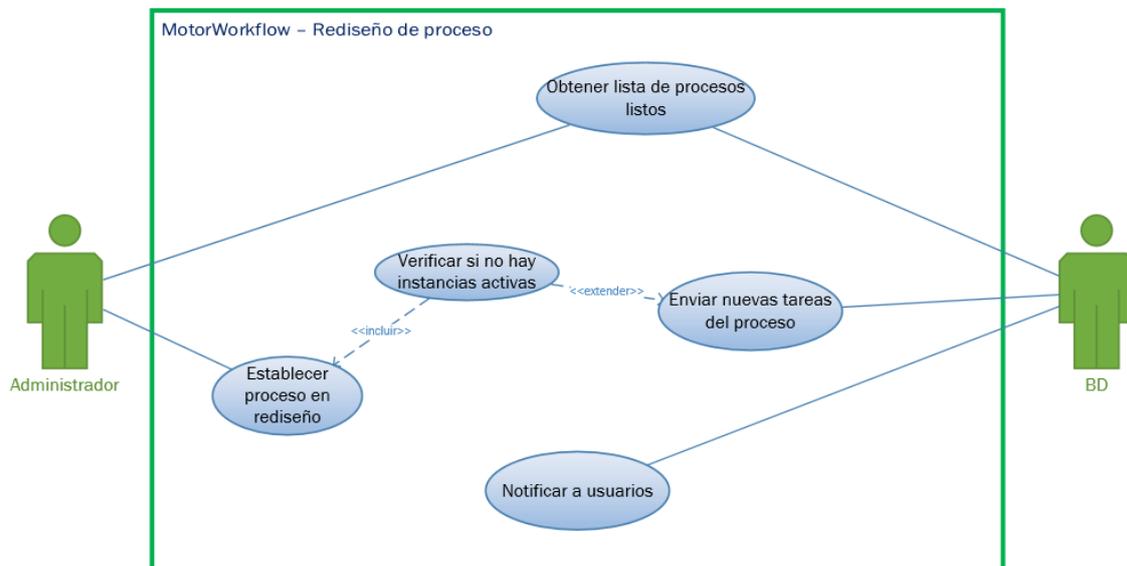


Figura 18. Diagrama UML del RF16  
Elaboración: Autores

### 1.17. DIAGRAMA 17: SUPERVISIÓN DE TAREAS (TRÁMITE) DEL PROCESO

Estará a cargo del supervisor del proceso, quien podrá visualizar la lista de tramites de cada proceso que tenga a su cargo, aquí se podrá realizar un cambio en la fecha de entrega, de igual forma se puede reasignar al usuario encargado de realizar la tarea.

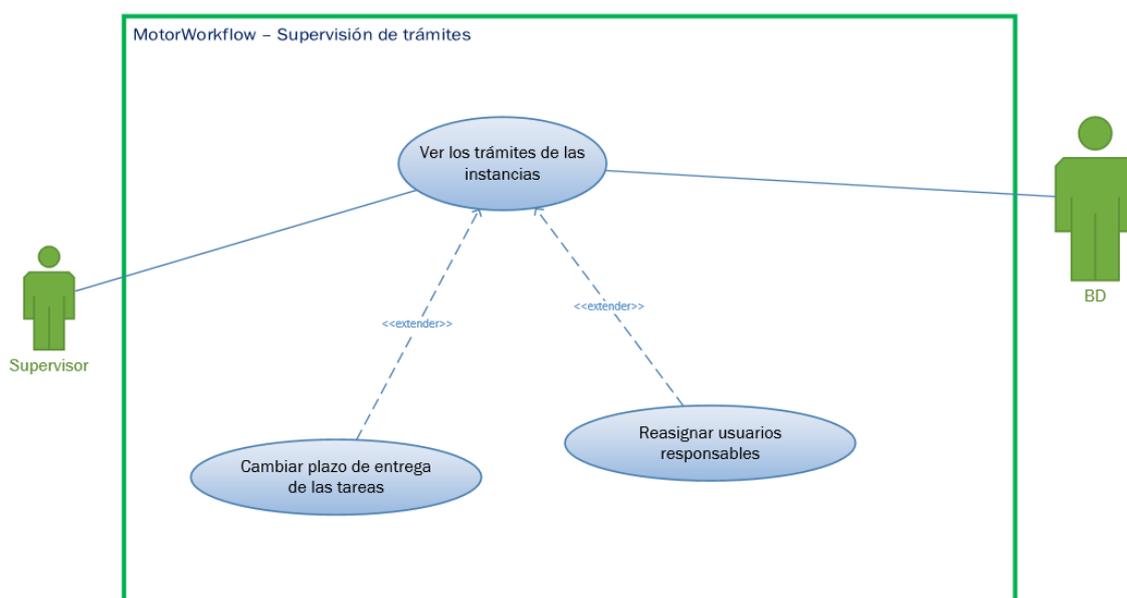


Figura 19. Diagrama UML del RF17  
Elaboración: Autores

### 1.18. DIAGRAMA 18: EJECUCIÓN DE LAS ACTIVIDADES

La Figura 20 muestra la manera en que un usuario con rol operador desarrolla una tarea.

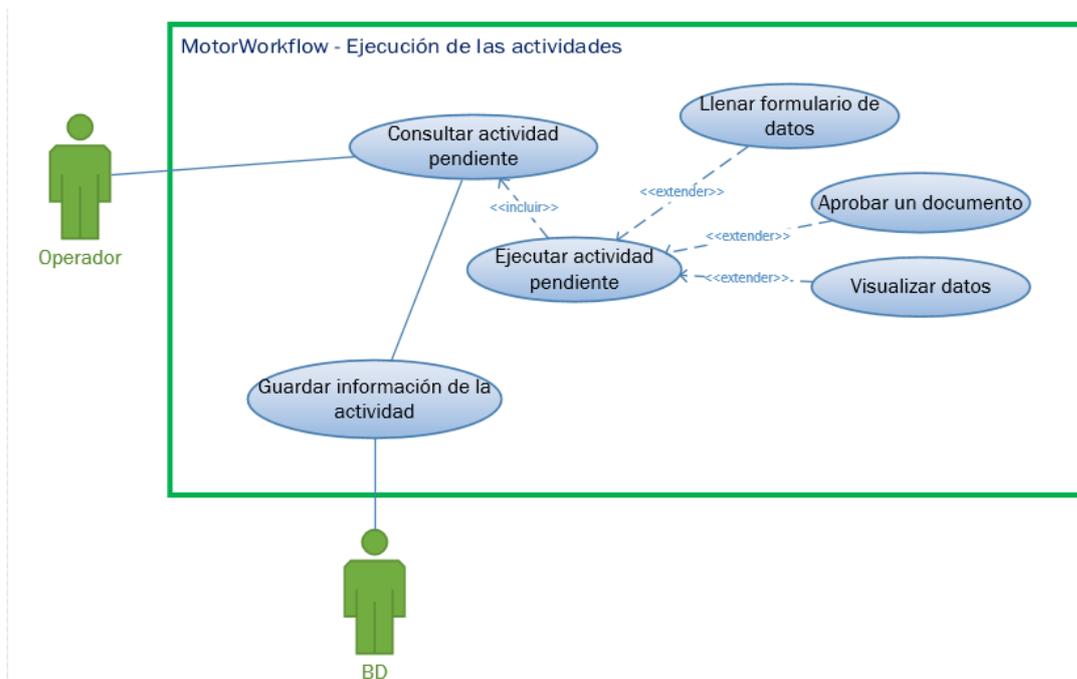


Figura 20. Diagrama UML del RF18  
Elaboración: Autores

### 1.19. DIAGRAMA 19: REINICIAR ACTIVIDAD

Si ocurre un error durante la ejecución de una actividad, se notificará del mismo al supervisor, quien podrá reiniciar la misma, como se muestra en la Figura 21

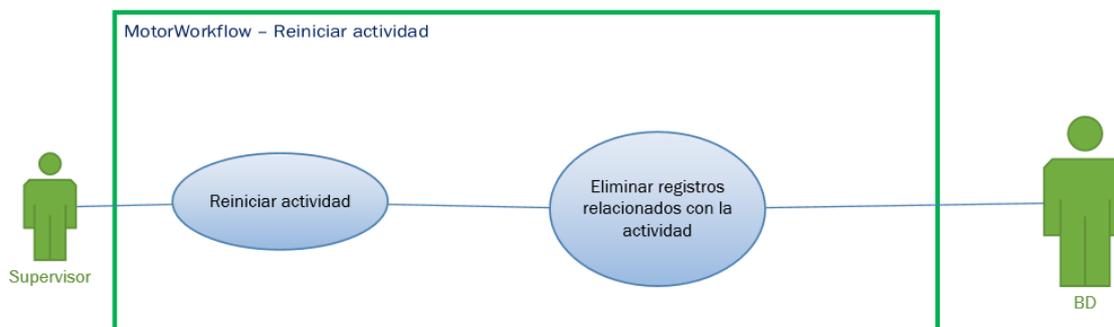


Figura 21. Diagrama UML del RF19  
Elaboración: Autores

## 1.20. DIAGRAMA 20: GESTIÓN DE CARGOS ADMINISTRATIVOS

Los cargos administrativos comprenden los cargos que podrán tener los usuarios del sistema tales como: gerente, secretaria, asistente, entre otros. Esta gestión comprende la creación, modificación y eliminación de dichos cargos.

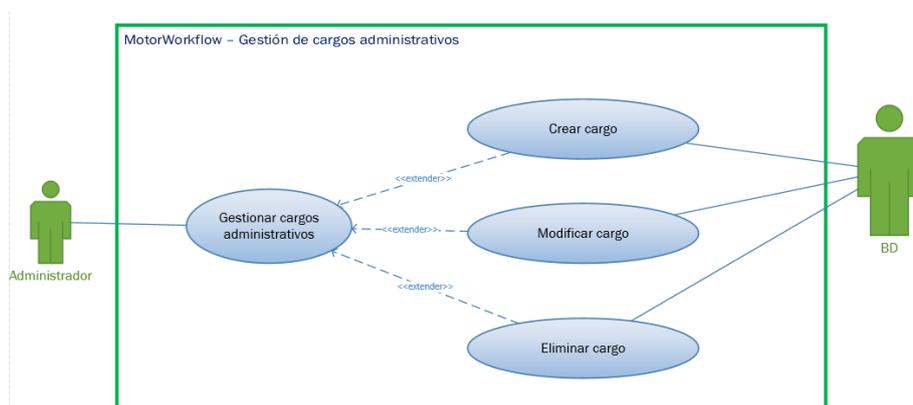


Figura 22. Diagrama UML del RF20  
Elaboración: Autores

## 1.21. DIAGRAMA 21: DEFINICIÓN DE PARTICIPANTES

El usuario con rol administrador realiza la asignación de participantes u operadores a una determinada tarea, esto corresponde a la asignación de recursos, sin embargo cuando se asigna como responsable un cargo administrativo el motor se debe encargar de gestionar o establecer quienes serán los usuarios participantes de esa actividad.

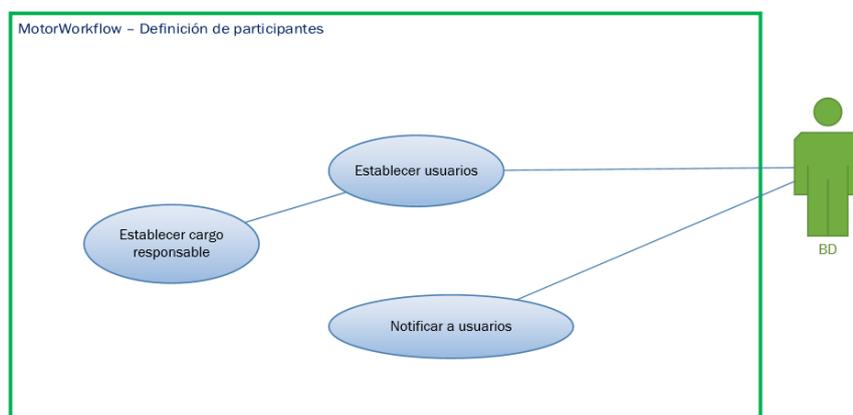


Figura 23. Diagrama UML del RF21  
Elaboración: Autores

## 1.22. DIAGRAMA 22: REASIGNACIÓN DE PARTICIPANTES

La reasignación de participantes consiste en cambiar de usuarios responsables de una tarea, esta reasignación la pueden realizar los usuarios con rol de administrador (para las tareas de definición) y supervisor (para los trámites).

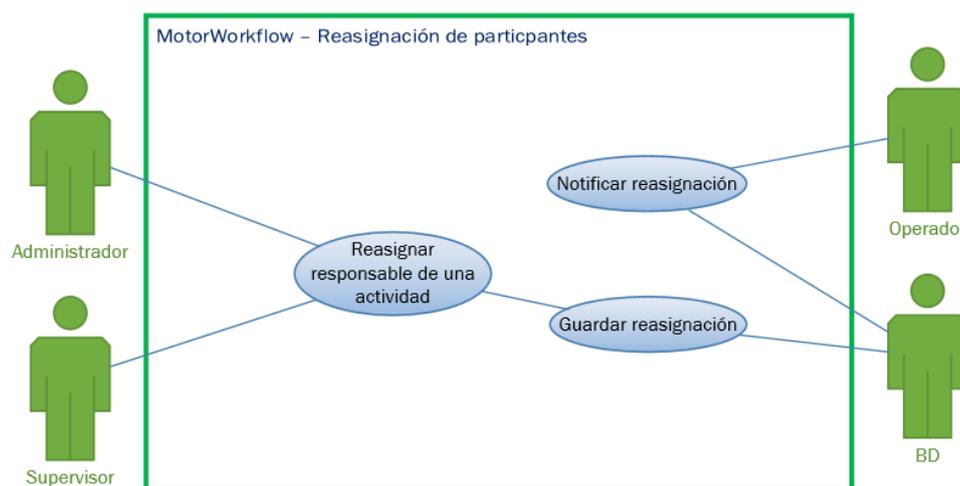


Figura 24. Diagrama UML del RF22.

Elaboración: Autores

## 1.23. DIAGRAMA 23: MECANISMOS DE CONTROL

El núcleo del Worklow realiza de manera interna y constante la revisión de los plazos de las actividades que se estén desarrollando, además se enviaran notificaciones o mensajes a los usuarios responsables de esa tarea, aquí no se presenta un diagrama ya que este mecanismo se encuentra implementado en muchas de los requerimientos mencionados anteriormente, como es el caso de la definición de tareas del proceso, en el cual se envían mensajes a los usuarios. Además de las notificaciones que se presentan en el diagrama que muestra la Figura 25

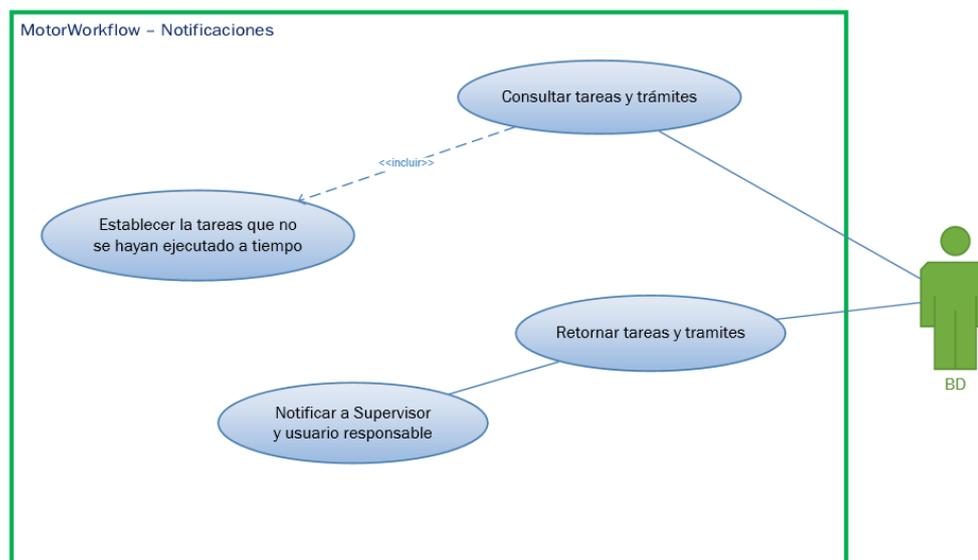
## 1.24. DIAGRAMA 24: ERRORES DURANTE LA EJECUCIÓN DE UN PROCESO

Constituye una forma de control en la ejecución de un proceso, no existe un diagrama específico ya que los errores que se especifican en este

requerimiento conllevan a tomar una decisión como las presentadas en las figuras 13, 14, 19 y 22.

### 1.25. DIAGRAMA 25: NOTIFICACIONES

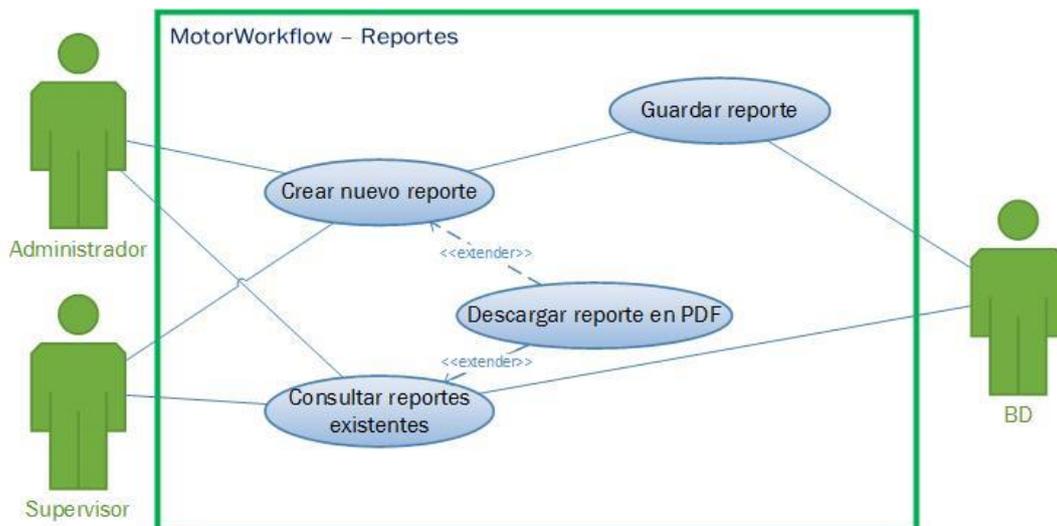
Cuando el núcleo encuentra un trámite que no se ha desarrollado en el plazo establecido envía un mensaje de alerta a los usuarios operador (de esa tarea) y supervisor (del proceso), además verifica el cumplimiento de las fechas límite en las tareas de definición enviando notificaciones al administrador y usuario responsable.



**Figura 25.** Diagrama UML del RF25  
Elaboración: Autores

### 1.26. DIAGRAMA 26: REPORTE

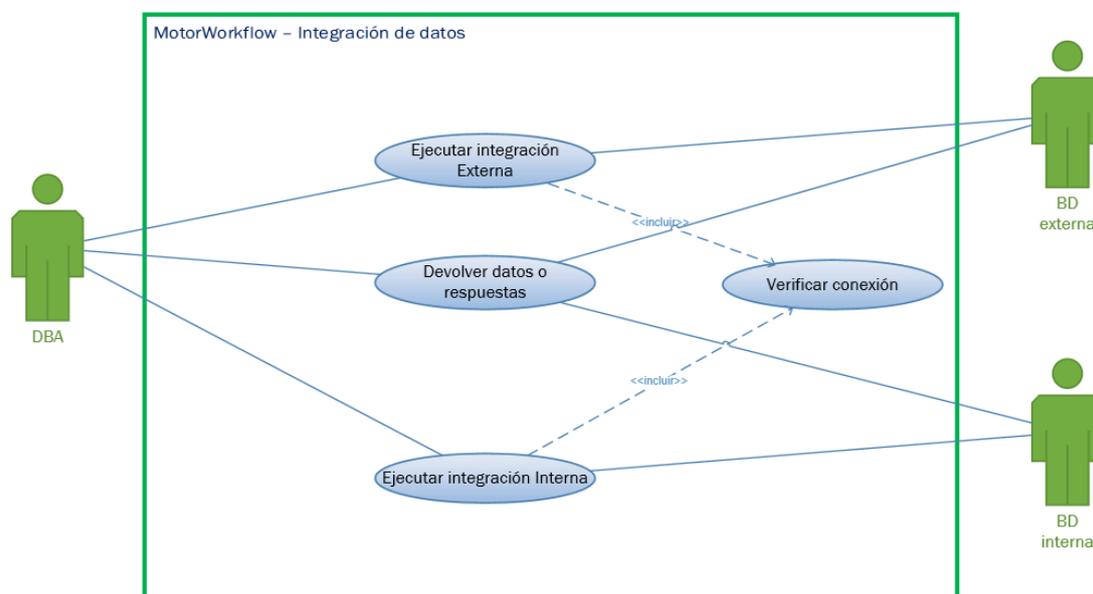
El núcleo permite al usuario administrador y supervisor, la realización de reportes estadísticos referente a los tiempos utilizados en cumplir una tarea, realizar un proceso, entre otros aspectos.



**Figura 26.** Diagrama UML del RF26  
**Elaboración:** Autores

### 1.27. DIAGRAMA 27: INTEGRACIÓN DE DATOS

La integración de datos consiste en la realización de las integraciones (tareas de definición) realizadas por el DBA, que ya han sido detalladas anteriormente.



**Figura 27.** Diagrama UML del RF27  
**Elaboración:** Autores

## ANEXO 4 – ELEMENTOS BPMN 2.0



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

# BPMN

## Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL



**BPMN 2.0:  
ELEMENTOS Y  
DOCUMENTACIÓN**

Dayana H. Bailón Delgado  
José S. Mera Cantos

**2017**

## FICHA DEL DOCUMENTO

El cuadro 1 detalla las versiones que ha tenido este anexo, indicando: fecha, versión, autores y por quien fue revisado.

**Cuadro 51.** Ficha de revisión del documento.

<b>Fecha</b>	<b>Versión</b>	<b>Autores</b>	<b>Verificado por.</b>
<b>06/10/2016</b>	2.0	Dayana Helen Bailón Delgado Karla Estefanía Cevallos Molina José Simón Mera Cantos Jeniffer Raquel Murillo Montesdeoca	Ing. Joffre Moreira Pico

**Elaboración:** Autores

## 1. ELEMENTOS BPMN 2.0 INCLUIDOS EN EL WORKFLOW

En el cuadro 2 se muestran los elementos de flujo BPMN 2.0 para la elaboración de diagramas de proceso simple, que se encuentran incluidos en la herramienta de modelado de las interfaces del Workflow y que pueden ser interpretados por el núcleo de la plataforma para su ejecución. Cada elemento cuenta con su descripción, clasificación y notación específica; además de los requerimientos que son necesarios para su ejecución en la columna denominada “Asignación de recursos”.

Cuadro 52. Elementos BPMN 2.0 incluidos en la herramienta de modelado de las interfaces del Workflow y que son interpretados por el núcleo de la plataforma para su ejecución.

ELEMENTO BÁSICO	DESCRIPCIÓN	ELEMENTOS ESPECÍFICOS	DESCRIPCIÓN	CLASIFICACIÓN			
				ELEMENTO ESPECIFICO	DESCRIPCIÓN	NOTACIÓN	ASIGNACIÓN DE RECURSOS
Event (Evento)	Describe algo que sucede durante el curso de un proceso. Afectan el flujo del modelo y usualmente tiene una causa (trigger, capturado) o un impacto (result, lanzado).	Eventos de Inicio para procesos de alto nivel. Actúa como disparador de un proceso.	Permite “capturar”. Representa el inicio de un proceso.	None (Ninguna)	No tiene establecida una condición o requisito para dar inicio al proceso o subprocesso.		<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Cargo</li> </ul>
				Message (Mensaje)	Un proceso o aplicativo recibe un mensaje específico para dar inicio a un proceso.		<ul style="list-style-type: none"> <li>• Código(s) de procesos que finalizan con un evento de fin de mensaje</li> </ul>
				None (Ninguna)	No tiene un resultado definido.		<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
		End (Fin)	Indica el fin de un proceso. Permite “lanzar”.	Message (Mensaje)	Este tipo de fin indica que se envía un mensaje a un participante al finalizar el proceso.		<ul style="list-style-type: none"> <li>• Códigos de proceso que tiene un evento de inicio de mensaje.</li> <li>• Mensaje.</li> <li>• Usuario</li> <li>• Cargo</li> <li>• Usuario instancia</li> </ul>

				Error	Indica que se debe generar un Error con nombre. Todos los subprocesos actualmente activos en el subproceso particular se terminan como resultado.		<ul style="list-style-type: none"> <li>Ninguno</li> </ul>
				Terminate (Terminar)	Este tipo de Fin indica que todas las Actividades en el Proceso deben ser terminadas inmediatamente.		<ul style="list-style-type: none"> <li>Ninguno</li> </ul>
<b>Gateway (Compuerta)</b>	Se determinará la ramificación, bifurcación, la fusión y la unión de caminos. Marcadores internos indicarán el tipo de control de la conducta.	Exclusive (Exclusiva)	Se utiliza para tomar una decisión entre dos opciones, basado en los datos que se envíen. Adicionalmente puede agregarse una opción por defecto en caso de que ninguna de las opciones esté acorde con la respuesta.	Divergencia	Aparecen dos o más rutas alternativas y solo una de ellas es válida en un momento determinado. La decisión depende del usuario. Una entrada. Dos o tres transiciones (salidas) de condición. (Tienen asociada una regla de negocio) Si no se asegura de que se escoja al menos una salida, es necesario ubicar un default.		Usuario: <ul style="list-style-type: none"> <li>Usuario responsable.</li> <li>Tiempo</li> <li>Vista de datos</li> <li>Usuario instancia</li> </ul>
							Regla: <ul style="list-style-type: none"> <li>Reglas por cada salida de la compuerta (operadores lógicos "=", "&lt;&gt;", "&gt;", "&lt;", "&gt;=", "&lt;=")</li> <li>Parámetro de la regla</li> <li>Dato de entrada de la regla</li> </ul>
				Convergencia	Dos o tres entradas. Una salida. Default no es necesario.		<ul style="list-style-type: none"> <li>Ninguno</li> </ul>
				Inclusive (Inclusiva)	Permite que se habiliten de dos a tres opciones que pueden ser tomadas en el flujo, dependiendo de la condición, es decir que los caminos no son excluyentes y se puede seguir ninguno, uno o más de los 3 disponibles. Pero el flujo debe diseñarse de	Divergencia	Basado en una compuerta exclusiva o datos de control del flujo de trabajo, uno o varios caminos son activados. Una entrada, dos o tres transiciones (salidas) de condición. (Tienen asociada una regla de negocio). Si no se asegura de que se escoja al menos una salida, es necesario

tal forma que se tome al menos un camino. Una compuerta inclusiva debe ser definida como elemento divergente para activar los caminos salientes y otra como elemento convergente para sincronizar.

ubicar un default.

Convergencia

Al menos un camino debió ser activado, por lo cual al menos un camino será sincronizado. Cuando la figura se utiliza como sincronizadora solo una transición sale de ésta. Dos o tres entradas. Una salida. Default no es necesario

- Ninguno

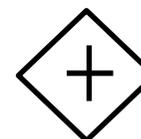
Parallel  
(Paralelo)

Permite habilitar dos o tres caminos de forma simultánea. Si se ubican tales caminos como salida, al ejecutarse la compuerta, automáticamente se van a disparar esos tres caminos, pero, si se ubican como entrada, deberán completarse tales flujos

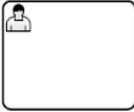
Divergencia

Indica puntos del proceso en la cual las actividades pueden ser llevadas a cabo en forma concurrente (paralela). Una entrada. Dos o tres salidas. No requiere ni reglas de negocio ni campo default, debido a que todas las transiciones que salen de esta figura se cumplen siempre.

- Ninguno



			<p>entrantes, para que pueda habilitarse la compuerta. Se recomienda el uso una Compuerta Paralelas utilizada como elemento convergente después de una o más Compuertas Paralelas como elementos divergentes, para poder sincronizar los caminos abiertos.</p>	<p>Convergencia</p>	<p>Sincroniza los caminos que parten de una Compuerta Paralela de divergencia, la compuerta paralela de convergencia permite que el flujo continúe, sólo hasta cuando todas las transiciones indicadas hayan llegado a la figura. Dos o tres entradas. Una salida. Default no es necesario</p>	<p>• Ninguno</p>
<p><b>Sequence Flow (Secuencia de flujo)</b></p>	<p>Se utiliza un flujo de secuencia para mostrar el orden de los elementos de flujo en un proceso. Cada flujo de secuencia tiene sólo una fuente y sólo un objetivo.</p>	<p>Sequence Flow (Secuencia de flujo)</p>	<p>Un flujo de secuencia permite la conexión entre el conjunto de los elementos de flujo como: Eventos, Actividades, Actividades Coreográficas y Gateways.</p>	<p>Sequence Flow (Secuencia de flujo)</p>	<p>Es la línea con una punta de flecha sólida que debe ser dibujado con una sola línea sólida para conectar elementos.</p>	<p> • Ninguno</p>
				<p>Default Sequence Flow (Secuencia de flujo predeterminada)</p>	<p>El flujo de secuencia predeterminado se toma (sólo se pasa un token) si todos los otros flujos de secuencia salientes de la actividad o puerta de enlace no son válidos (sus condiciones son falsas).</p>	<p> • Ninguno</p>
				<p>Data Association (Asociación de datos)</p>	<p>Una asociación se utiliza para asociar información y artefactos con objetos de flujo. Los objetos de flujo y no gráficos de texto pueden asociarse con los objetos de flujo y el flujo.</p>	<p> • Ninguno</p>
				<p>Directional Association (Asociación Direccional)</p>	<p>Se utilizan para mostrar cómo Data Objects eran entradas o salidas a actividades. Las asociaciones de datos se utilizan</p>	<p>• Ninguno</p>

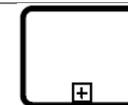
					para mover datos entre objetos de datos, propiedades y entradas y salidas de actividades, procesos y GlobalTasks.		
<b>DataObject (Objeto de datos)</b>	Representan un documento, un dato o un conjunto de los mismos.	DataObject (Objeto de datos)	Los objetos de datos representan los datos y documentos en un proceso, suelen definir las entradas y salidas de las actividades	Data Object (Objeto de datos)	Los elementos de objeto de datos pueden hacer referencia opcionalmente a un elemento DataState, que es el estado de los datos contenidos en el objeto de datos.		<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Data Stores (Almacenes de datos)</b>	Mecanismo para recuperar información.			Data Stores (Almacenes de datos)	El mismo almacén de datos se puede visualizar, a través de una referencia. Recupera datos almacenados que persistirá más allá del alcance del proceso		<ul style="list-style-type: none"> <li>• Ninguno</li> </ul>
<b>Activity (Actividad)</b>	Describe el tipo de trabajo que será realizado.  Denomina una actividad que será realizada por un participante del proceso.  Una actividad puede ser en función de su ejecución (manual o automática)	Task (Tarea)	Se considera como actividad atómica.  Representa una sola unidad de trabajo que no es o no se puede dividir a un mayor nivel de detalle de procesos de negocio.	None (Nada o simple)	Una tarea se utiliza cuando el trabajo en el proceso no se puede desglosar hasta un nivel de detalle más fino.		<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Usuario instancia</li> <li>• Tiempo</li> <li>• Tarea</li> </ul>
				Service Task (Tarea de Servicio)	Utiliza algún tipo de servicio, que podría ser un servicio Web o una aplicación automatizada.		<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Cargo</li> <li>• Usuario instancia</li> <li>• Reporte</li> </ul>
				User Task (Tarea de usuario)	Tarea típica de "flujo de trabajo" donde un intérprete humano realiza la tarea con la ayuda de una aplicación de software. Las Tareas de Usuario de BPMN deben ser renderizadas en interfaces de usuario como clientes de formularios, etc.		<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Usuario instancia</li> <li>• Tiempo</li> <li>• Tarea</li> </ul>
				Manual Task (Tarea manual)	Una tarea manual es una tarea que se espera que se realice sin la ayuda de cualquier motor de ejecución de procesos de negocio o cualquier aplicación.		<ul style="list-style-type: none"> <li>• Usuario</li> <li>• Usuario instancia</li> <li>• Tiempo</li> <li>• Tarea</li> </ul>

---

Process/Sub  
Process  
(Subproceso)

Call Activity (Actividad de llamada)

Una actividad de llamada identifica un punto en el proceso en el que se utiliza un proceso global o una tarea global. Se considera como actividad no atómica.



- Código único del proceso al que se realiza la llamada.

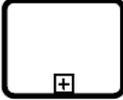
---

**Elaboración:** Autores

## 2. REGLAS DE CONEXIÓN DE FLUJO DE SECUENCIA

En el siguiente cuadro muestra los objetos de flujo BPMN 2.0 y cómo estos objetos pueden conectarse entre sí a través de secuencias (indica que el objeto enumerado en la fila puede conectarse al objeto enumerado en la columna). Sólo se muestran en la tabla aquellos objetos que pueden tener flujos de secuencia entrantes y / o salientes. Por lo tanto, objeto de datos, almacén de datos y anotación de texto no se enumeran en la tabla.

**Cuadro 53-** Reglas de conexión de flujo de secuencia

ELEMENTO (Para toda su clasificación)	ENTRADAS	SALIDAS	CONECTA		NO CONECTA	
			ENTRADA	SALIDA	ENTRADA	SALIDA
	0	1	-	 	-	 
	1 o mas	0	 	-	 	-
	1 o mas	1	  	  -	 -	  -
	1 o mas	1	  	  		
	1 – 3 (para mayor presión en los resultados)	1 – 3 (La salida por defecto no es necesaria en todas las compuestas)	  	  		

Elaboración: Autores

## ANEXO 5 – PLANIFICACIÓN DE INCREMENTOS



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

**MEIDAW**

## Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL



**INCREMENTOS  
PLANIFICADOS  
CONTROL E INFORMES**

Dayana H. Bailón Delgado  
José S. Mera Cantos

**2017**

## FICHA DEL DOCUMENTO

El cuadro 1 se detalla las versiones que ha tenido este anexo, indicando: fecha, versión, autores y por quien fue revisado.

**Cuadro 54.** Ficha del documento.

<b>Fecha</b>	<b>Versión</b>	<b>Autores</b>	<b>Verificado por</b>
27/01/2017	1.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira

**Elaboración.** Autores.

## **CONTENIDO DE CUADROS Y FIGURAS**

Cuadro 1. Ficha del documento. ....	197
Cuadro 2. Planificación de incrementos 001 .....	200
Cuadro 3. Planificación de incrementos 002 .....	201
Cuadro 4. Planificación de incrementos 003 .....	203
Cuadro 5. Planificación de incrementos 004 .....	204
Cuadro 6. Planificación de incrementos 005 .....	205
Cuadro 7. Planificación de incrementos 006 .....	206
Cuadro 8. Planificación de incrementos 007 .....	207
Cuadro 9. Planificación de incrementos 008 .....	208
Cuadro 10. Planificación de incrementos 009 .....	209
Cuadro 11. Planificación de incrementos 010 .....	211
Cuadro 12. Planificación de incrementos 011 .....	212
Cuadro 13. Planificación de incrementos 012 .....	214
Cuadro 14. Planificación de incrementos 013 .....	215
Cuadro 15. Planificación de incrementos 014 .....	216

## PLANIFICACIÓN DE INCREMENTOS

Esta planificación de Incrementos está basada de acuerdo a los requerimientos funcionales definidos en la especificación de requerimientos de software IEE830 (Anexo 2), teniendo en cuenta las actividades programadas para alcanzar el desarrollo de dichos requerimientos, los incrementos del núcleo del Workflow se han programado para realizarse cada 15 días, al finalizar estos días se comprueba los avances realizados, mediante los siguiente puntos:

- **Planificación:** se especifican las actividades a realizar los próximos 15 días (lunes-viernes), y la administración de los recursos de manera cronológica, considerando:
  - ✓ Asignaciones
  - ✓ Notificaciones
  - ✓ Permisos
- **Ejecución:** Desarrollo de planificación.
- **Control de los incrementos:** Conformado por:
  - ✓ Feedbacks positivos: Desvíos favorables del sistema
  - ✓ Feedbacks negativos: Fallas del sistema
- **Informes de avance:** se indica:
  - ✓ Estado actual del sistema.
  - ✓ Actividades completadas.
  - ✓ Actividades retrasadas

## 1. INCREMENTO 001

Corresponde a la planificación para la instalación de las herramientas para el desarrollo del núcleo, tal como se indica en el cuadro 2. Además se inició con el diseño de la base de datos, así como el análisis de los elementos a utilizar para el modelamiento del diagrama (procedimiento) de los procesos.

**Cuadro 55.** Planificación de incrementos 001

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 05/12/2016</b>		<b>Fecha Fin: 23/12/2016</b>		<b>Id. 001</b>
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Instalar y configurar Visual Studio 2015.	Dayana Bailón	Ninguno	Ninguno
2	Creación del proyecto en Asp.net Web api	José Mera	Ninguno	Ninguno
3	Publicar el proyecto en team foundation	Jose Mera	Incluir a las autoras de la tesis complementaria	Ninguno
4	Instalar y configurar SQL Server 2008 R2.	José Mera	Ninguno	Ninguno
5	Diseñar el esquema de la Base de Datos	Dayana Bailón José Mera	Ninguno	Ninguno
6	Análisis de elementos BPMN 2.0 a utilizar	Dayana Bailón José Mera	Ninguno	Ninguno
<b>EJECUCIÓN</b>				
<b>Terminado</b>				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS</b>	<b>POSITIVOS</b>	Ninguno		
	<b>NEGATIVOS</b>	Ninguno		
<b>INFORMES DE AVANCES</b>				
<b>Estado Actual del Sistema</b>	<b>n. de actividad</b>	<b>Completada</b>	<b>Retrasada</b>	
<b>Primera planificación, el sistema se encuentra en estado de inicio. Se analizaron los elementos del BPMN para guardarlos en la BD.</b>	1	✓	-	
	2	✓	-	
	3	✓	-	
	4	✓	-	
	5	✓	-	
	6	✓	-	
<b>Elaboración. Autores</b>				

## 2. INCREMENTO 002

En el desarrollo de este incremento realiza la implementación del modelo inicial de la base de datos (este modelo puede cambiar con el paso del tiempo), al SGBD escogido, para de este modo iniciar con la programación sobre el lenguaje seleccionado, creando la capa de acceso a los datos.

Por otra parte se empezó con el desarrollo de los store procedure que gestionan la inserción o modificación de los datos enviados desde las interfaces a los recursos, de igual forma los que se encargarán de seleccionar los datos requeridos o de eliminar aquellos que el usuario indique a través de los recursos web api.

En este incremento se desarrollan los RF 1, 2 y 20 establecidos en el Anexo 2.

**Cuadro 56.** Planificación de incrementos 002

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 26/12/2016</b>		<b>Fecha Fin: 13/01/2017</b>		<b>Id. 002</b>
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Implementar el esquema en SQL SERVER 2008.	Dayana Bailón	Ninguno	Ninguno
2	Desarrollar Store Procedure (CRUD) de las tablas iniciales, mismas que serán utilizadas de manera inicial en el desarrollo de las interfaces.	Dayana Bailón José Mera	Ninguno	Ninguno
3	Crear capa de acceso a la BD	José Mera	Ninguno	Ninguno
4	Desarrollar recursos CRUD para cargos.	José Mera	Notificar la realización de la URI	Ninguno
5	Desarrollar recursos CRUD para Usuarios.	Dayana Bailón	Notificar la realización de la URI	Ninguno
6	Desarrollar recursos CRUD para Acceso.	José Mera	Notificar la realización de la URI	Ninguno
7	Desarrollo de store procedure para la definición de tareas de un proceso.	Dayana Bailón	Ninguno	Ninguno

8	Desarrollo de métodos lógicos para la definición de tareas de un proceso.	Dayana Bailón	Ninguno	Ninguno
9	Desarrollo de recursos para la definición de tareas de un proceso.	Dayana Bailón	Notificar la realización de la URI	Ninguno
10	Desarrollo de store procedure para la supervisión de tareas de un proceso.	José Mera	Ninguno	Ninguno
11	Desarrollo de métodos lógicos para la supervisión de tareas de un proceso.	José Mera	Ninguno	Ninguno

## EJECUCIÓN

Terminado

## CONTROL DE INCREMENTOS

<b>FEEDBACKS POSITIVOS</b>	Implementación de las buenas prácticas para realizar web api
<b>NEGATIVOS</b>	Ninguno

## INFORMES DE AVANCES

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
	1	✓	-
El desarrollo del núcleo está en su parte medular, como es la base de datos y el desarrollo de store procedure. Además ya se pueden crear las tareas que se necesitan para la implementación del proceso, con la notificación respectiva al supervisor del proceso.	2	✓	-
	3	✓	-
	4	✓	-
	5	✓	-
	6	✓	-
	7	✓	-
	8	✓	-
	9	✓	-
	10	✓	-
	11	✓	-

Elaboración. Autores

### 3. INCREMENTO 003

En este incremento se culmina con el desarrollo de las clases entidades con sus respectivas variables y las lógicas que contienen los métodos necesarios para el correcto funcionamiento del núcleo.

Además, se implementaron diversas librerías que permitieron al análisis del diagrama del proceso para su validación y posteriormente guardarlo en la base de datos y de igual forma para los formularios creados.

En este incremento se desarrollan los RF 3 y 4 especificados en el anexo 2.

Cuadro 57. Planificación de incrementos 003

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 16/01/2017</b>		<b>Fecha Fin: 03/02/2017</b>		<b>Id. 003</b>
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Desarrollar clases entidades restantes.	Dayana Bailón	Ninguno	Ninguno
2	Desarrollar clases lógicas restantes.	José Mera	Ninguno	Ninguno
3	Desarrollo de store procedure para guardar las tareas del proceso sin finalizar y finalizadas.	José Mera	Ninguno	Ninguno
4	Desarrollo de métodos para la validación del diagrama.	Dayana Bailón	Ninguno	Ninguno
5	Desarrollo de método para guardar las tareas del proceso sin finalizar.	José Mera	Ninguno	Ninguno
6	Desarrollo de método para guardar tareas finalizadas	José Mera	Ninguno	Ninguno
7	Crear recurso para validar y guardar el diagrama del proceso.	Dayana Bailón	Notificar la realización de la URI	Ninguno
8	Desarrollar recurso para validar y guardar la tarea de formulario.	José Mera	Notificar la realización de la URI	Ninguno
9	Desarrollo de store procedure que seleccione los mensajes de las tareas de los usuarios.	José Mera	Ninguno	Ninguno
10	Desarrollar métodos para el retorno de los mensajes a la interfaz.	José Mera	Ninguno	Ninguno
11	Desarrollo de recurso que retorna los mensajes de las tareas al usuario que inicia sesión.	José Mera	Notificar la realización de la URI	Ninguno
12	Mejoras al método de guardar tareas de definición de procesos.	Dayana Bailón	Notificar cambios	Ninguno
<b>EJECUCIÓN</b>				
<b>Terminado</b>				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS</b>	<b>POSITIVOS</b>	Implementación de nuevas librerías para el análisis del diagrama del proceso y de los formularios. Se realizaron cambios en los recursos con el fin de que estos retornen los mensajes, para notificar a los usuarios correspondientes.		
	<b>NEGATIVOS</b>	Ninguno		
<b>INFORMES DE AVANCES</b>				

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
	1	✓	-
	2	✓	-
	3	✓	-
El desarrollo del núcleo avanza de manera satisfactoria, de acuerdo a lo establecido en el cronograma del proyecto. Ya se pueden guardar los formularios y el diagrama del proceso. Se retornan los mensajes para las notificaciones a los usuarios conectados.	4	✓	-
	5	✓	-
	6	✓	-
	7	✓	-
	8	✓	-
	9	✓	-
	10	✓	-
	11	✓	-
	12	✓	-

Elaboración. Autores

#### 4. INCREMENTO 004

Incremento dedicado para la ejecución de la integración interna, la obtención de los datos de los formularios, las validaciones y la selección de dichos datos de acuerdo al tipo de filtro utilizado.

De igual forma se procede a la realización del controlador para la verificación de los servicios web, la implementación de una nueva clase para realizar la validación y poder retornar los campos de entrada y salida que tiene el servicio.

El desarrollo de este incremento se relaciona con los RF 5 y 6 del anexo 2.

**Cuadro 58.** Planificación de incrementos 004

PLANIFICACIÓN DE INCREMENTOS				
Fecha Inicio: 06/02/2017		Fecha Fin: 24/02/2017		Id. 004
Objetivo a largo plazo		Elaborar el núcleo con base a los requerimientos estipulados		
PLANIFICACIÓN				
n.	Actividades	Asignaciones	Notificaciones	Permisos
1	Desarrollo de store procedure para realizar la integración interna	Dayana Bailón	Ninguno	Ninguno
2	Desarrollo de métodos verificación y ejecución de una integración interna.	Dayana Bailón	Ninguno	Ninguno

3	Desarrollo de recurso para la integración interna	Dayana Bailón	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para realizar una integración Externa	José Mera	Ninguno	Ninguno
5	Desarrollo de métodos para la ejecución de una integración externa.	José Mera	Ninguno	Ninguno
6	Desarrollo de recursos para la verificación de la integración externa	José Mera	Notificar la realización de la URI	Ninguno

**EJECUCIÓN**

Terminado

**CONTROL DE INCREMENTOS**

<b>FEEDBACKS POSITIVOS</b>	Implementación de nuevas librerías que permiten el análisis de un servicio web.
<b>NEGATIVOS</b>	Ninguno.

**INFORMES DE AVANCES**

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El desarrollo del núcleo avanza, se cuenta con la validación de las integraciones internas, obteniendo los datos registrados por medio de los formularios. Además se pueden implementar los servicios web	1	✓	-
	2	✓	-
	3	✓	-
	4	✓	-
	5	✓	-
	6	✓	-

Elaboración. Autores

**5. INCREMENTO 005**

Incremento dedicado para la ejecución de las integraciones (interna y externa), obteniendo los datos, para luego ser presentados en una vista de datos o en un reporte. Esto corresponde a los RF 7 y 8 del anexo 3.

Cuadro 59. Planificación de incrementos 005

PLANIFICACIÓN DE INCREMENTOS				
Fecha Inicio: 27/02/2017		Fecha Fin: 17/03/2017		Id. 005
<b>Objetivo a largo plazo</b>	Elaborar el núcleo con base a los requerimientos estipulados			
PLANIFICACIÓN				
n.	Actividades	Asignaciones	Notificaciones	Permisos
1	Desarrollo de store procedure para realizar una vista de datos.	José Mera	Ninguno	Ninguno

2	Desarrollo de métodos necesarios para mostrar los datos de las integraciones.	José Mera	Ninguno	Ninguno
3	Desarrollo de recurso para la mostrar los datos de la integración.	José Mera	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para realizar un reporte.	Dayana Bailón	Ninguno	Ninguno

**EJECUCIÓN**

Terminado

**CONTROL DE INCREMENTOS**

FEEDBACKS POSITIVOS Ninguno

NEGATIVOS Ninguno

**INFORMES DE AVANCES**

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El núcleo puede obtener la información de las integraciones para ser presentadas a través de una vista de datos o un formulario.	1	✓	-
	2	✓	-
	3	✓	-
	4	✓	-

Elaboración. Autores

**6. INCREMENTO 006**

En este incremento se desarrollaron varios recursos: para la revisión de tareas, para obtener los datos de la tarea, además de los métodos en la lógica que son necesarios, ya que se debe notificar al usuario responsable si la tarea fue aprobada o no. El incremento corresponde al RF 9 del anexo 3.

Cuadro 60. Planificación de incrementos 006

PLANIFICACIÓN DE INCREMENTOS				
Fecha Inicio: 20/03/2017		Fecha Fin: 07/04/2017		Id. 006
Objetivo a largo plazo		Elaborar el núcleo con base a los requerimientos estipulados		
PLANIFICACIÓN				
n.	Actividades	Asignaciones	Notificaciones	Permisos
1	Desarrollo de store procedure para realizar la revisión de tareas.	José Mera	Ninguno	Ninguno
2	Desarrollo de métodos necesarios para la revisión de las tareas.	José Mera	Ninguno	Ninguno

3	Desarrollo de recursos para la revisión de las tareas.	José Mera	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para obtener los datos de las tareas.	Dayana Bailón	Ninguno	Ninguno
5	Desarrollo de métodos necesarios para obtener los datos de las tareas.	Dayana Bailón	Ninguno	Ninguno
6	Desarrollar recurso para obtener datos de las tareas.	Dayana Bailón	Notificar la realización de la URI	Ninguno

**EJECUCIÓN**

Terminado

**CONTROL DE INCREMENTOS**

FEEDBACKS POSITIVOS Ninguno

NEGATIVOS Ninguno

**INFORMES DE AVANCES**

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El incremento permite que el núcleo pueda mostrar los datos de las tareas, para que el administrado procesa a su revisión y según su criterio aprobarla o rechazarla.	1	✓	-
	2	✓	-
	3	✓	-
	4	✓	-
	5	✓	-
	6	✓	-

Elaboración. Autores

**7. INCREMENTO 007**

Implementación del paquete Swagger para realizar la documentación de los recursos (URI) del núcleo del workflow. En cuestión de seguridad se implementó la autenticación OAuth 2.0 y la autorización para usuarios autenticados.

Cuadro 61. Planificación de incrementos 007

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
Fecha Inicio: 10/04/2017		Fecha Fin: 28/04/2017		Id. 007
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
n.	Actividades	Asignaciones	Notificaciones	Permisos
1	Descarga del paquete Swagger.	Dayana Bailón	Ninguno	Ninguno

2	Implementación de Swagger.	Dayana Bailón	Ninguno	Ninguno
3	Implementación de la autenticación OAuth 2.0.	José Mera	Notificación sobre la restricción a los recursos.	Ninguno

**EJECUCIÓN**

Terminado

**CONTROL DE INCREMENTOS**

<b>FEEDBACKS POSITIVOS</b>	Se utilizó la autenticación OAuth 2.0 basada en tokens, para prevenir el acceso de usuarios desconocidos a los recursos.
<b>NEGATIVOS</b>	No se pudo implementar la autenticación bajo el formato de Swagger.

**INFORMES DE AVANCES**

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El núcleo cuenta con seguridad basada en token y con el acceso a los recursos solo por parte de usuarios autenticados.	1	✓	-
	2	✓	-
	3	✓	-

Elaboración. Autores

**8. INCREMENTO 008**

Se estableció la forma de realizar los reportes que serán generados por el sistema, además se realizaron varios recursos para la generación del reporte del proceso en la etapa de diseño.

Además, durante este incremento se realizó la reasignación de fechas de entrega y de usuarios responsables de las tareas de definición de los procesos. En este incremento se realiza el RF 26 y 22 del anexo 2.

**Cuadro 62.** Planificación de incrementos 008

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 01/05/2017</b>		<b>Fecha Fin: 19/05/2017</b>		<b>Id. 008</b>
<b>Objetivo a largo plazo</b>	Elaborar el núcleo con base a los requerimientos estipulados			
<b>PLANIFICACIÓN</b>				
n.	Actividades	Asignaciones	Notificaciones	Permisos
1	Desarrollo de store procedure para guardar datos de los reportes generados.	José Mera	Ninguno	Ninguno

2	Desarrollo de métodos para la generación de los reportes.	José Mera	Ninguno	Ninguno
3	Desarrollo del recurso para guardar el registro del reporte generado.	José Mera	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para la reasignación de fecha y usuario de una tarea.	Dayana Bailón	Ninguno	Ninguno
5	Método para reasignación de fechas y usuarios de una tarea	Dayana Bailón	Ninguno	Ninguno
6	Desarrollo de recurso para la reasignación de fecha y usuario de una tarea.	Dayana Bailón	Notificar la realización de la URI	Ninguno

**EJECUCIÓN**

Terminado

**CONTROL DE INCREMENTOS**

FEEDBACKS POSITIVOS Ninguno

NEGATIVOS Ninguno

**INFORMES DE AVANCES**

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El núcleo puede gestionar los datos para realizar los reportes necesarios y guardar un registro del mismo. Además se crea el recurso para la reasignación de los usuarios a las tareas.	1	✓	-
	2	✓	-
	3	✓	-
	4	✓	-
	5	✓	-
	6	✓	-

Elaboración. Autores

**9. INCREMENTO 009**

Se establecen los recursos necesarios para la estación del supervisor, tales como los gráficos estadísticos, la lista de los procesos a los que tiene que supervisar así como la lista de las instancias del mismo.

Cuadro 63. Planificación de incrementos 009

PLANIFICACIÓN DE INCREMENTOS				
Fecha Inicio: 22/05/2017	Fecha Fin: 09/06/2017	Id. 009		
Objetivo a largo plazo	Elaborar el núcleo con base a los requerimientos estipulados			
PLANIFICACIÓN				
n.	Actividades	Asignaciones	Notificaciones	Permisos

1	Desarrollo de store procedure para obtener los datos de los proceso.	Dayana Bailón	Ninguno	Ninguno
2	Desarrollo de los métodos que muestran la información de los procesos.	Dayana Bailón	Ninguno	Ninguno
3	Desarrollo de recursos para mostrar la información de los procesos que un supervisor tiene a su cargo.	José Mera	Notificar la realización de la URI	Ninguno

#### EJECUCIÓN

Terminado

#### CONTROL DE INCREMENTOS

FEEDBACKS POSITIVOS Ninguno

NEGATIVOS Ninguno

#### INFORMES DE AVANCES

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El núcleo puede gestionar los datos para que las interfaces los puedan mostrar al usuario correspondiente.	1	✓	-
	2	✓	-
	3	✓	-

Elaboración. Autores

## 10. INCREMENTO 010

En este incremento se realiza la asignación de recursos, para poder realizar esta asignación el motor debe haber identificado cada uno de los elementos que intervienen en el diagrama y haberlos guardado en la BD, una vez realizado esto se puede proceder a la asignación, la cual está dividida por elementos, cuando un elemento ha sido asignado, este se puede modificar. Además existe la validación en los elementos de compuerta (basados en reglas), para que antes de ser asignadas las reglas la actividad anterior deba tener lista la asignación.

El desarrollo de este incremento corresponde a los RF 10 y 11 del anexo 2.

**Cuadro 64.** Planificación de incrementos 010

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio:</b> 12/06/2017		<b>Fecha Fin:</b> 30/06/2017		<b>Id.</b> 010
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Desarrollo de store procedure para la definición del proceso.	José Mera	Ninguno	Ninguno
2	Desarrollo de los métodos que analizaran el modelo, identificando los elementos, para posteriormente guardarlos en la BD.	Dayana Bailón José Mera	Ninguno	Ninguno
3	Desarrollo de store procedure para la asignación de recursos.	José Mera	Ninguno	Ninguno
4	Métodos necesarios para realizar la asignación.	José Mera	Ninguno	Ninguno
5	Recursos para la asignación de atributos a los elementos.	Dayana Bailón	Notificar la realización de la URI	Ninguno
<b>EJECUCIÓN</b>				
Terminado				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS</b>	<b>POSITIVOS</b>	Ninguno		
	<b>NEGATIVOS</b>	Ninguno		
<b>INFORMES DE AVANCES</b>				
<b>Estado Actual del Sistema</b>	<b>n. de actividad</b>	<b>Completada</b>	<b>Retrasada</b>	
<b>El núcleo identifica los elementos del proceso y los guarda en la BD. Gestiona los diferentes recursos del proceso para realizar la asignación de los mismos a las actividades correspondientes.</b>	1	✓	-	
	2	✓	-	
	3	✓	-	
	4	✓	-	
	5	✓	-	
Elaboración. Autores				

## 11. INCREMENTO 011

En este incremento se inicia la ejecución de los procesos, además se implementa la autenticación OAuth 2.0 en Swagger.

Una vez realizada la asignación de recursos, el proceso estará listo para ser instanciado, en el incremento se especifican las diversas opciones que se pueden hacer en la instancia de un proceso.

Además, se desarrolla la ejecución de las actividades y su respectivo monitoreo por parte del supervisor del proceso. Se desarrollan los RF 12, 13, 14, 15, 17, 18 y 19 del anexo 2.

**Cuadro 65.** Planificación de incrementos 011

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 03/07/2017</b>		<b>Fecha Fin: 21/07/2017</b>		<b>Id. 011</b>
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Desarrollo de store procedure para iniciar la instancia de un proceso.	José Mera	Ninguno	Ninguno
2	Desarrollo de los métodos para iniciar una instancia.	José Mera	Ninguno	Ninguno
3	Desarrollo de recurso para iniciar una instancia	José Mera	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para eliminar una instancia	Dayana Bailón	Ninguno	Ninguno
5	Desarrollo de los métodos para eliminar una instancia.	Dayana Bailón	Ninguno	Ninguno
6	Desarrollo de recurso para eliminar una instancia	Dayana Bailón	Notificar la realización de la URI	Ninguno
7	Desarrollo de store procedure para reiniciar la instancia de un proceso.	Dayana Bailón	Ninguno	Ninguno
8	Desarrollo de los métodos para reiniciar una instancia.	Dayana Bailón	Ninguno	Ninguno
9	Desarrollo de recurso para reiniciar una instancia	Dayana Bailón	Notificar la realización de la URI	Ninguno
10	Implementar autenticación OAuth 2.0a swagger.	José Mera	Ninguno	Ninguno
11	Desarrollo de store procedure para la ejecución de las actividades (trámites).	José Mera	Ninguno	Ninguno
12	Desarrollo de métodos para la ejecución de los trámites.	José Mera	Ninguno	Ninguno
13	Desarrollo de recursos para la ejecución de los trámites	José Mera	Notificar la realización de la URI	Ninguno

14	Desarrollo de store procedure para reiniciar un trámite.	Dayana Bailón	Ninguno	Ninguno
15	Desarrollo de los métodos para reiniciar un trámite.	Dayana Bailón	Ninguno	Ninguno
16	Desarrollo de recurso para reiniciar un trámite.	Dayana Bailón	Notificar la realización de la URI	Ninguno
17	Desarrollo de métodos para supervisión de trámites	José Mera	Ninguno	Ninguno
18	Desarrollo de recursos para supervisión de los trámites.	José Mera	Notificar la realización de la URI	Ninguno

## EJECUCIÓN

Terminado

## CONTROL DE INCREMENTOS

**FEEDBACKS POSITIVOS** Se logra implementar la autenticación a Swagger

**NEGATIVOS** Ninguno

## INFORMES DE AVANCES

Estado Actual del Sistema	n. de actividad	Completada	Retrasada
El núcleo permite que se inicie la instancia de un proceso, además de que en caso de presentarse un error se pueda gestionar el mismo, dándole algunas opciones al supervisor del mismo. Se ejecutan las actividades, las cuales son monitoreadas por el supervisor, en caso de existir un error durante la ejecución de la misma, esta puede ser reiniciada.	1	✓	-
	2	✓	-
	3	✓	-
	4	✓	-
	5	✓	-
	6	✓	-
	7	✓	-
	8	✓	-
	9	✓	-
	10	✓	-
	11	✓	-
	12	✓	-
	13	✓	-
	14	✓	-
	15	✓	-
	16	✓	-
	17	✓	-
	18	✓	-

Elaboración. Autores

## 12. INCREMENTO 012

En este incremento se realiza el rediseño de los procesos, así como los diferentes métodos (automáticos) de verificación del cumplimiento de las tareas de definición de procesos y de los trámites que se encuentren en ejecución.

Luego se pasa a la fase de pruebas, para lo cual se sigue el plan de pruebas especificado en el anexo 6.

Los RF desarrollados son los 16 y 23, además en este incremento se incluye la planificación de pruebas y la ejecución de pruebas funcionales.

**Cuadro 66.** Planificación de incrementos 012

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 24/07/2017</b>		<b>Fecha Fin: 11/08/2017</b>		<b>Id. 012</b>
<b>Objetivo a largo plazo</b>		Elaborar el núcleo con base a los requerimientos estipulados Ejecutar pruebas de funcionamiento del núcleo		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Desarrollo de store procedure para el rediseño del proceso	Dayana Bailón	Ninguno	Ninguno
2	Métodos necesarios para el rediseño del proceso	Dayana Bailón	Ninguno	Ninguno
3	Recurso para el rediseño del proceso.	Dayana Bailón	Notificar la realización de la URI	Ninguno
4	Desarrollo de store procedure para la verificación de las tareas de definición del proceso	José Mera	Ninguno	Ninguno
5	Desarrollo de store procedure para la verificación de los trámites.	José Mera	Ninguno	Ninguno
6	Desarrollo de Jobs para la verificación de las tareas y los trámites.	José Mera	Ninguno	Ninguno
7	Desarrollo del plan de pruebas	José Mera	Ninguno	Ninguno
8	Ejecución de pruebas funcionales	José Mera Dayana Bailón	Ninguno	Ninguno
<b>EJECUCIÓN</b>				
Terminado				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS</b>	<b>POSITIVOS</b>	Se implementó el Sql Agent para la ejecución de los Jobs		
	<b>NEGATIVOS</b>	Ninguno		
<b>INFORMES DE AVANCES</b>				
<b>Estado Actual del Sistema</b>	<b>n. de actividad</b>	<b>Completada</b>	<b>Retrasada</b>	
El núcleo puede realizar el rediseño de los procesos.	1	✓	-	
Se permite la verificación de los plazos de entrega tanto de las tareas como de los trámites mediante Jobs de SQL.	2	✓	-	
Se inició la ejecución de las pruebas para verificar el funcionamiento del motor.	3	✓	-	
	4	✓	-	
	5	✓	-	
	6	✓	-	
	7	✓	-	
	8	✓	-	
Elaboración. Autores				

### 13. INCREMENTO 013

En este se continúa con la ejecución de pruebas, esta vez corresponde a la realización de las correcciones de los errores encontrados cuando se realizaron las pruebas. Además se realiza el manual del programador.

**Cuadro 67.** Planificación de incrementos 013

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio:</b> 14/08/2017		<b>Fecha Fin:</b> 31/08/2017		<b>Id.</b> 013
<b>Objetivo a largo plazo</b>		Ejecutar pruebas de funcionamiento del núcleo		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Corregir los errores documentados	Dayana Bailón José Mera	Ninguno	Ninguno
2	Desarrollo del manual de programador	Dayana Bailón	Ninguno	Ninguno
<b>EJECUCIÓN</b>				
Terminado				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS POSITIVOS</b>	Se implementó el Sql Agent para la ejecución de los Jobs			
<b>NEGATIVOS</b>	Ninguno			
<b>INFORMES DE AVANCES</b>				
<b>Estado Actual del Sistema</b>	<b>n. de actividad</b>	<b>Completada</b>	<b>Retrasada</b>	
Se realizaron las pruebas funcionales.	1	✓	-	
	2	✓	-	
Elaboración. Autores				

### 14. INCREMENTO 014

Este incremento consiste en la integración del motor con las interfaces de la tesis complementaria, su implementación en los servidores de la institución y la verificación del correcto funcionamiento del sistema.

Cuadro 68. Planificación de incrementos 014

<b>PLANIFICACIÓN DE INCREMENTOS</b>				
<b>Fecha Inicio: 04/09/2017</b>		<b>Fecha Fin: 22/09/2017</b>		<b>Id. 014</b>
<b>Objetivo a largo plazo</b>		Implementar el núcleo en los servidores de la Institución		
<b>PLANIFICACIÓN</b>				
<b>n.</b>	<b>Actividades</b>	<b>Asignaciones</b>	<b>Notificaciones</b>	<b>Permisos</b>
1	Dar acceso a los recursos web, para usuarios autenticados.	José Mera	Ninguno	Ninguno
2	Instalar el sistema en el servidor.	Dayana Bailón José Mera	Ninguno	Ninguno
3	Verificar el correcto funcionamiento del sistema.	Dayana Bailón José Mera	Ninguno	Ninguno
<b>EJECUCIÓN</b>				
Terminado				
<b>CONTROL DE INCREMENTOS</b>				
<b>FEEDBACKS POSITIVOS</b>	Ninguno			
<b>NEGATIVOS</b>	Ninguno			
<b>INFORMES DE AVANCES</b>				
<b>Estado Actual del Sistema</b>	<b>n. de actividad</b>	<b>Completada</b>	<b>Retrasada</b>	
<b>Se integra el sistema y se implementa en los servidores de la institución.</b>	1	✓	-	
	2	✓	-	
	3	✓	-	
Elaboración. Autores				

## ANEXO 6 – PLAN DE PRUEBAS



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

**MEIDAW**

**Núcleo de un Workflow de  
Procesos Administrativos de la ESPAM MFL**



**PLAN DE PRUEBAS**

Dayana H. Bailón Delgado  
José S. Mera Cantos

**2017**

## FICHA DEL DOCUMENTO

El Cuadro 69 detalla las versiones que ha tenido este anexo, indicando: fecha, versión, autores y por quien fue revisado.

**Cuadro 69.** Ficha del documento.

<b>Fecha</b>	<b>Versión</b>	<b>Autores</b>	<b>Verificado por</b>
<b>24/02/2017</b>	1.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira

**Elaboración.** Autores

## CONTENIDO GENERAL

FICHA DEL DOCUMENTO .....	218
CONTENIDO GENERAL.....	219
CONTENIDO DE CUADROS Y FIGURAS.....	220
1. INTRODUCCIÓN.....	221
1.1. PROPÓSITO.....	221
1.2. OBJETIVOS.....	221
1.3. REFERENCIAS Y DOCUMENTOS FUENTE .....	221
1.4. FUNCIONES DEL SISTEMA .....	222
1.4.1. INCLUSIONES .....	222
1.4.2. EXCLUSIONES.....	222
1.5. ÁMBITO DE PRUEBA.....	223
1.5.1. PRUEBAS DE UNIDAD.....	223
1.5.2. PRUEBAS DE REGRESIÓN .....	223
1.5.3. PRUEBAS DE FUNCIONALIDAD .....	223
1.6. RECURSOS.....	224
1.6.1. HUMANOS .....	224
1.6.2. HARDWARE.....	224
1.6.3. SOFTWARE .....	224
1.7. MÉTODO DE REALIZACIÓN DE PRUEBAS.....	224
2. AUTOMATIZACIÓN PROCESOS .....	225
2.1.1. CASOS DE PRUEBAS .....	225

## CONTENIDO DE CUADROS Y FIGURAS

<b>Cuadro 1.</b> Ficha del documento.....	218
<b>Cuadro 2:</b> Especificación de las pruebas de unidad.....	223
<b>Cuadro 3:</b> Especificación de las pruebas de unidad.....	223
<b>Cuadro 4:</b> Especificación de las pruebas de unidad.....	224
<b>Cuadro 5.</b> Caso de prueba del diseño del proceso (DP). ....	225
<b>Cuadro 6.</b> Caso de prueba de supervisión del proceso (SP).....	226
<b>Cuadro 7.</b> Caso de prueba de realizar tarea (RT). ....	226
<b>Cuadro 8.</b> Caso de prueba de aprobar o rechazar (AR).....	227
<b>Cuadro 9.</b> Caso de prueba de rediseño de proceso (RP).....	227
<b>Cuadro 10.</b> Caso de prueba de definición del proceso (DP). ....	228
<b>Cuadro 11.</b> Caso de prueba de asignación de recursos al diagrama (ARD). ....	228
<b>Cuadro 12.</b> Caso de prueba de iniciar instancia (II).....	229
<b>Cuadro 13.</b> Caso de prueba de ejecutar actividad (EA). ....	229
<b>Cuadro 14.</b> Caso de prueba de supervisión de trámites (ST).....	230
<b>Cuadro 15.</b> Caso de prueba de mecanismos de control (MC).....	230
<b>Cuadro 16.</b> Caso de prueba de funcionalidad. ....	232
<b>Cuadro 17.</b> Caso de prueba funcional en la supervisión de procesos.....	233

## **8. INTRODUCCIÓN**

Este documento contiene la planificación para el desarrollo de un proceso, mediante la herramienta desarrollada a la que denominaremos “Workflow”. La automatización de un proceso mediante el Workflow consta de tres etapas: El diseño del proceso, la asignación de recursos y la ejecución del proceso.

### **8.1. PROPÓSITO**

Describir la forma en que se realizarán las pruebas del Workflow para la creación, ejecución y documentación de los procesos administrativos de la ESPAM MFL.

El desarrollo de las pruebas consta de tres etapas principales:

- Enfoque de la prueba: Establece el alcance de las pruebas del sistema, la estrategia general que se adopta, las actividades a realizar, los recursos generales necesarios y los métodos y procesos que se utilizarán.
- Planificación de la prueba: Detalla las actividades, las dependencias y el esfuerzo necesarios para llevar a cabo la prueba del sistema.
- Condiciones y Casos de prueba: Documentación de las pruebas que se aplican, los datos a ser procesados y los resultados esperados.

### **8.2. OBJETIVOS**

Este diseño de pruebas tiene como objetivo demostrar las siguientes especificaciones:

- Las funcionalidades realizadas en el desarrollo, son las que se indican en el documento de especificación de requerimientos Anexo 2.
- El software mejora las funciones de negocio previstas y alcanza los estándares requeridos para la sistematización de los procesos.

### **8.3. REFERENCIAS Y DOCUMENTOS FUENTE**

- Especificación de Requerimientos Funcionales.
- Diagrama de Transición de Estados.
- Modelo Entidad - Relación.

- Diseño del Workflow.

## **8.4. FUNCIONES DEL SISTEMA**

### **INCLUSIONES**

- Definición de tareas para la creación de un proceso.
- Supervisión de tareas para diseño del proceso.
- Definición del Flujo (diagrama).
- Diseñador de Formularios.
- Integración Interna (II).
- Integración Externa (IE).
- Vista de datos.
- Reporte de formularios.
- Aprobar o rechazar tareas durante el diseño del proceso.
- Definición de un proceso.
- Asignación de recursos al diagrama.
- Instancia de un proceso.
- Eliminar instancia de un proceso.
- Reiniciar instancia de un proceso.
- Historial de instancia de un proceso.
- Rediseño de un proceso.
- Supervisión de tareas (trámite) del proceso.
- Ejecución de las actividades.
- Reiniciar actividad.
- Definición de participantes.
- Reasignación de participantes.
- Mecanismos de Control.
- Errores durante la ejecución de un proceso.
- Notificaciones.
- Reportes.
- Integración de datos.

### **EXCLUSIONES**

- Login del Usuario.

- Gestión de cargos administrativos.

## 8.5. ÁMBITO DE PRUEBA

A continuación, se describen los tipos de pruebas principales que se llevarán a cabo para verificar el correcto funcionamiento del Workflow. Todos los planes de prueba del sistema y las condiciones se desarrollan a partir de la especificación funcional y especificaciones de procesos del Workflow.

### PRUEBAS DE UNIDAD

Consisten en la verificación del funcionamiento de cada requerimiento, teniendo en cuenta cada una de las especificaciones necesarias para completar cada funcionalidad.

**Cuadro 70:** Especificación de las pruebas de unidad.

<b>OBJETIVO:</b>	<b>Asegurar que la información fluye de forma adecuada desde y hacia el recurso del motor que está siendo probado.</b>
<b>TÉCNICAS:</b>	Se utiliza la herramienta swagger para realizar cada caso de prueba.
<b>CRITERIOS DE FINALIZACIÓN:</b>	Se ha probado todos los recursos que son utilizados en el funcionamiento del Workflow y evaluado todas las condiciones lógicas del mismo.
<b>CONSIDERACIONES ESPECIALES:</b>	Se prueban únicamente funcionalidades específicas en cada una de las etapas del desarrollo de un proceso (diseño, asignación y ejecución).
<b>Elaboración:</b> Autores	

### PRUEBAS DE REGRESIÓN

Consisten en la verificación de los errores encontrados durante la ejecución de pruebas unitarias.

**Cuadro 71:** Especificación de las pruebas de unidad.

<b>OBJETIVO:</b>	<b>Comprobar las correcciones determinadas en los recursos probados, garantizando la estabilidad del motor.</b>
<b>TÉCNICAS:</b>	Se utiliza la herramienta swagger para realizar cada prueba de regresión.
<b>CRITERIOS DE FINALIZACIÓN:</b>	Se ha probado todos los recursos que son utilizados y evaluado todas las condiciones especificadas.
<b>CONSIDERACIONES ESPECIALES:</b>	Se prueban únicamente funcionalidades en las que se hayan detectado algún error o cuyo resultado no sea el esperado.
<b>Elaboración:</b> Autores	

### PRUEBAS DE FUNCIONALIDAD

Consisten en la verificación del funcionamiento de cada requerimiento, mediante el uso de las interfaces (aplicación cliente).

**Cuadro 72:** Especificación de las pruebas de unidad.

<b>OBJETIVO:</b>	<b>Asegurar que cada elemento cumple con los requerimientos indicados.</b>
	Se realizan pruebas alfa que cubren los siguientes aspectos:
<b>TÉCNICAS:</b>	<b>Pruebas de especificaciones funcionales:</b> Se trata de pruebas de bajo nivel que tienen como objetivo poner a prueba los procesos individuales y los flujos de datos. Se crean casos de prueba para recorrer todos los posibles flujos de la aplicación y determinar si el flujo de la información es correcto y completo (Prueba alfa).
<b>CRITERIOS DE FINALIZACIÓN:</b>	Se ha realizado las pruebas alfa.
<b>CONSIDERACIONES ESPECIALES:</b>	Se realizarán combinaciones con el fin de que cada requerimiento del motor del workflow sea probado.
<b>Elaboración:</b> Autores	

## 8.6. RECURSOS

### HUMANOS

- 1 Tester.
- 5 usuarios uno con un perfil (rol) de sistema (Administrador, Supervisor, Modelador, DBA y, Diseñador) diferente.
- 3 usuarios con rol operador con cargos administrativos diferentes.

### HARDWARE

- 8 PC, Procesador de 1.70 GHz o más, Memoria RAM de 4 MB o más, Espacio en Disco Duro de 500 GB o más.
- 1 PC, Procesador de 2.53 GHz, Memoria RAM de 6 MB o más, Espacio en Disco Duro de 2 TB.

### SOFTWARE

- Sistema Operativo Windows XP, Vista, 7, 8 o versiones superiores.
- Browser de cualquier tipo.
- Sistema Operativo Windows Server 2008 R2.

## 8.7. MÉTODO DE REALIZACIÓN DE PRUEBAS

Las pruebas de unidad serán realizadas al finalizar el desarrollo de cada requerimiento por la persona que lo programa.

Las pruebas de funcionalidad del sistema serán realizadas por el tester en un inicio, una vez todos los errores detectados hayan sido depurados el tester realizará la certificación del sistema con 8 usuarios (con rol de sistema y operadores) que harán uso del Workflow.

## 9. AUTOMATIZACIÓN PROCESOS

### CASOS DE PRUEBAS

- **Pruebas Unitarias.**

**Componente unitario:** Diseño del proceso (requerimiento 1).

**Precondición:** Debe existir al menos un usuario con todos los roles del sistema.

**Cuadro 73.** Caso de prueba del diseño del proceso (DP).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_DP_01	Buscar usuarios con rol específico que se encuentren activos.	Listado de los usuarios activos correspondientes al rol enviado.	Lista con todos los usuarios activos, que tengan asignado el rol especificado.	Correcto
PU_DP_02	Probar que sucede si el título o el código del proceso ya existen en la BD.	Retorna lista de mensajes vacía.	La lista de mensajes retorna vacía.	Correcto
PU_DP_03	Probar que sucede al guardar un proceso	Retorna lista de mensajes para los usuarios responsables de las tareas.	Lista de mensajes para los usuarios.	Correcto

Elaboración. Autores.

**Componente Unitario:** Supervisión de tareas de diseño del proceso (requerimiento 2).

**Precondición:** Debe existir al menos un proceso en etapa de diseño.

**Cuadro 74.** Caso de prueba de supervisión del proceso (SP).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_SP_01	Enviar lista de tareas de un proceso, con todos sus elementos.	Retorna objeto de cabecera con la lista de tareas y sus respectivos elementos.	Cabecera del proceso con lista de tareas.	Correcto
PU_SP_02	Probar que sucede si una tarea no es ejecutada en el plazo establecido.	Lista de mensajes, con el mensaje de la tarea atrasada en alerta.	Mensaje de alerta.	Correcto
PU_SP_03	Probar que sucede si se modifica la fecha límite del proceso.	Retorna lista de mensajes con la fecha actualizada del proceso.	Mensaje con la nueva fecha del proceso.	Correcto
PU_SP_04	Probar que sucede si se modifica la fecha límite y/o el responsable de una tarea.	Retorna lista de mensajes con la actualización de la tarea.	Mensaje con la nueva fecha y/o responsable de la tarea.	Correcto

Elaboración. Autores.

**Componente Unitario:** Recurso para guardar una tarea (para continuar con su realización después) y guardar enviándole una notificación al administrador del proceso (requerimientos del 3 al 8).

**Precondición:** Debe existir al menos una tarea de cada tipo (Formulario, Reporte de Formulario, Vista de datos Formulario, Integración Interna, Integración Externa, Vista de datos integración, Reporte de integración y Diagrama).

**Cuadro 75.** Caso de prueba de realizar tarea (RT).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_RT_01	Probar que sucede si el modelo no cumple las especificaciones de conexión indicadas en el anexo 4.	Lista de elementos del diagrama, con su respectivo estado y mensaje de error.	Lista de elementos con su estado y el mensaje que especifica el error cometido.	Correcto
PU_RT_02	Probar que sucede si el formulario tiene errores.	Retorna estado booleano false.	Estado false	Correcto
PU_RT_03	Obtener formularios aprobados para realizar Integraciones Internas, vista de datos o reportes de formulario.	Listado de formularios aprobados.	Lista de formularios aprobados.	Correcto

PU_RT_04	Obtener parámetros de integración externa.	Lista de parámetros de entrada y salida.	Lista de parámetros.	Correcto
PU_RT_05	Obtener integraciones internas o externas para vista de datos o reportes de integración.	Listado de integraciones internas o externas aprobadas.	Lista de integraciones aprobadas.	Correcto
PU_RT_06	Probar que sucede si una tarea es guardada.	Listado de elementos si la tarea es un diagrama, caso contrario la lista estará vacía.	Listado de elementos del diagrama. Lista vacía para el resto de tareas.	Correcto
PU_RT_07	Probar que sucede si una tarea es guardada y enviada al administrador.	Listado de mensajes.	Mensaje al administrador para que revise la tarea. Mensaje al usuario indicándole que la tarea fue enviada.	Correcto

Elaboración. Autores.

**Componente Unitario:** Aprobar o rechazar tareas (requerimiento 9).

**Precondición:** Debe existir al menos una tarea de cada tipo (Formulario, Reporte de Formulario, Vista de datos Formulario, Integración Interna, Integración Externa, Vista de datos integración, Reporte de integración y Diagrama), enviada para su revisión al administrador.

**Cuadro 76.** Caso de prueba de aprobar o rechazar (AR).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_AR_01	Probar que sucede si la tarea se aprueba.	Retorna mensaje de aprobación.	Mensaje que indica al usuario que la tarea ha sido aprobada	Correcto
PU_AR_02	Probar que sucede si la tarea se rechaza.	Retorna mensaje de rechazo.	Mensaje al usuario que indica el motivo por el cual la tarea fue rechazada.	Correcto

Elaboración. Autores.

**Componente Unitario:** Rediseño de proceso (requerimiento 16).

**Precondición:** Debe existir al menos un proceso instanciado.

**Cuadro 77.** Caso de prueba de rediseño de proceso (RP).

CÓDIGO	DESCRIPCIÓN	RESULTADO	RESULTADO	ESTADO
--------	-------------	-----------	-----------	--------

		ESPERADO	OBTENIDO	
PU_RP_01	Probar que sucede si el proceso no está listo para rediseño.	Valor booleano false, que indica la existencia de instancias en ejecución.	Retorna false.	Correcto
PU_RP_02	Probar que sucede si el proceso está listo para rediseño.	Valor booleano true, que indica que no hay instancias en ejecución.	Retorna true.	Correcto

Elaboración. Autores.

### Componente Unitario: Definición del proceso (requerimiento 10)

**Precondición:** Debe existir al menos un proceso terminado (progreso al 100%).

**Cuadro 78.** Caso de prueba de definición del proceso (DP).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_DP_01	Probar que sucede si el proceso no está al 100%	Retorna objeto de proceso vacío.	Objeto proceso vacío.	Correcto
PU_DP_02	Probar que sucede si el proceso está terminado.	Retorna objeto de proceso, con la lista de los elementos que intervienen en el diagrama del flujo.	Objeto proceso con lista de elementos.	Correcto

Elaboración. Autores.

### Componente Unitario: Asignación de recursos al diagrama (requerimiento 11)

**Cuadro 79.** Caso de prueba de asignación de recursos al diagrama (ARD).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_ARD_01	Obtener lista de atributos por cada elemento del diagrama.	Retorna lista de atributos de los elementos.	Lista de atributos de los elementos.	Correcto
PU_ARD_02	Obtener recursos disponibles para asignar al diagrama.	Retorna lista de recursos disponibles.	Lista de recursos disponibles.	Correcto

Elaboración. Autores

**Componente Unitario:** Iniciar Instancia (requerimiento 12)

**Precondición:** Debe existir al menos un proceso listo para ser instanciado, es decir que ya se le realizó la asignación de recursos.

**Cuadro 80.** Caso de prueba de iniciar instancia (II).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_II_01	Probar que sucede cuando se inicia una instancia.	Listado de mensajes.	Mensaje al supervisor indicándole el inicio de una nueva instancia. Mensaje al operador de la actividad.	Correcto

Elaboración. Autores

**Componente Unitario:** Ejecutar actividad (requerimiento 12)**Cuadro 81.** Caso de prueba de ejecutar actividad (EA).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_EA_01	Probar que sucede si la actividad es una compuerta paralela.	Retorna mensajes de las siguientes actividades	Mensajes para los usuarios responsables de las siguientes actividades.	Correcto
PU_EA_02	Probar que sucede si la actividad es una compuerta exclusiva por decisión.	Mensaje para el usuario encargado de tomar la decisión.	Mensaje para el usuario encargado de tomar la decisión.	Correcto
PU_EA_03	Probar que sucede si la actividad es una compuerta exclusiva por regla.	Ejecución automática, se notifica al usuario de la siguiente tarea.	Ejecución automática, se notifica al usuario de la siguiente tarea.	Correcto
PU_EA_04	Probar que sucede si la actividad es una compuerta inclusiva.	Ejecución automática, se notifica a e o los usuario de las actividades que se activen.	Ejecución automática, se notifica a e o los usuario de las actividades que se activen.	Correcto
PU_EA_05	Probar que sucede si la actividad es una actividad de servicio.	Ejecución automática con notificación a usuarios encargados, se notifica de la siguiente actividad	Ejecución automática con notificación a usuarios encargados, se notifica de la siguiente actividad	Correcto
PU_EA_06	Probar que sucede	Notificación al	Notificación al	Correcto

	si la actividad es una actividad manual.	usuario responsable.	usuario responsable.	
PU_EA_07	Probar que sucede si la actividad es una actividad simple o de usuario.	Notificación al usuario responsable.	Notificación al usuario responsable.	Correcto

Elaboración. Autores

**Componente Unitario:** Supervisión de trámites de la instancia de un proceso (requerimiento 17).

**Precondición:** Debe existir un proceso en ejecución.

**Cuadro 82.** Caso de prueba de supervisión de trámites (ST).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
PU_ST_01	Enviar lista de trámites de una instancia, con todos sus elementos.	Retorna objeto de instancia con la lista de trámites y sus respectivos elementos.	Instancia con lista de trámites.	Correcto
PU_ST_02	Probar que sucede si un trámite no es ejecutado en el plazo establecido.	Lista de mensajes, con el mensaje de la tarea atrasada en alerta.	Mensaje de alerta para supervisor y operador.	Correcto
PU_ST_03	Probar que sucede si se modifica la fecha límite del trámite.	Retorna lista de mensajes con la fecha actualizada del proceso.	Mensaje con la nueva fecha del trámite.	Correcto
PU_ST_04	Probar que sucede si se modifica el responsable de un trámite.	Retorna lista de mensajes con la actualización de la tarea.	Mensaje para el nuevo responsable del trámite.	Correcto

Elaboración. Autores.

**Componente Unitario:** Mecanismos de control (requerimiento 23), aquí se prueban los siguientes mecanismos que se pueden utilizar durante la ejecución de un proceso, se incluyen los requerimientos 13, 14, 19 y 22. Al hacer uso de estos mecanismos se crea un registro en historial (requerimiento 15).

**Precondición:** Deben existir errores en la ejecución de una instancia, estos errores se detallan en el requerimiento 24.

**Cuadro 83.** Caso de prueba de mecanismos de control (MC).

CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
--------	-------------	--------------------	--------------------	--------

<b>PU_MC_01</b>	Probar que sucede si se elimina una instancia.	Eliminación de todos los registros correspondientes a la instancia indicada.	Registros de la instancia eliminados.	Correcto
<b>PU_MC_02</b>	Probar que sucede si se reinicia la instancia.	Se eliminan los registros de la instancia, pero vuelven a crearse los trámites.	Registros eliminados, se crean nuevamente los trámites del proceso.	Correcto
<b>PU_MC_03</b>	Probar que sucede si se reinicia una actividad.	Se eliminan los registros de la actividad y vuelve a ejecutarse.	Registros de la actividad eliminados, se vuelve a ejecutar la actividad.	Correcto
<b>PU_MC_04</b>	Probar que sucede si se reasigna el responsable de una actividad.	Mensaje al nuevo usuario.	Mensaje que indica la reasignación al nuevo usuario.	Correcto

Elaboración. Autores

- **Pruebas de Regresión.**

Al no existir errores durante la ejecución de las pruebas de unidad, no se consideró necesario ejecutar pruebas de regresión.

- **Pruebas de Funcionalidad.**

Estas pruebas se aplicaron luego de que el motor y las interfaces fueron implementados en el servidor de la UPS, es decir que dichas pruebas se realizaron mediante la aplicación cliente, para comprobar que los datos sean recibidos y enviados de forma correcta, siguiendo el flujo interfaz->motor->BD y viceversa, además de verificar la ejecución de actividades según el diagrama especificado, teniendo en cuenta los tiempos asignados.

En el Cuadro 84 se especifican cada una de las pruebas realizadas en cada etapa de la automatización del proceso.

**Cuadro 84.** Caso de prueba de funcionalidad.

ETAPA	CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
DISEÑO	PF_DP_01	Creación del proceso y todos sus elementos.	Notificación a los usuarios involucrados.	Notificaciones.	Correcto
	PF_DP_02	Revisar notificación de tarea.	Datos del proceso y de la tarea a realizar.	Datos de cabecera tarea.	Correcto
	PF_DP_03	Realizar diagrama.	Diagrama con los elementos indicando si es correcto o no, según las reglas de conexión.	Listado de elementos con sus estados.	Correcto
	PF_DP_04	Realizar Formulario.	Notificación si es correcto o no.	Notificación.	Correcto
	PF_DP_05	Realizar Integración Interna.	Lista de formularios Aprobados.	Lista de formularios.	Correcto
	PF_DP_06	Realizar Integración Externa.	Lista de parámetros.	Lista de parámetros.	Correcto
	PF_DP_07	Realizar Vista de Datos Integración	Lista de integraciones aprobadas.	Lista de integraciones.	Correcto
	PF_DP_08	Realizar Vista de Datos Formulario	Lista de formularios aprobados.	Lista de formularios.	Correcto
	PF_DP_09	Realizar Reporte Integración.	Lista de integraciones aprobadas.	Lista de integraciones.	Correcto
	PF_DP_10	Realizar Reporte Formulario.	Lista de formularios aprobados.	Lista de formularios.	Correcto
	PF_DP_11	Enviar Tarea	Notificación al administrador y usuario.	Notificaciones.	Correcto
	PF_DP_12	Aceptar/Rechazar tarea.	Notificación al usuario.	Notificación.	Correcto
	PF_DP_13	Definición del proceso.	Lista de elementos del diagrama.	Lista de elementos.	Correcto
	PF_DP_14	Asignar recursos a eventos.	Listado de recursos disponibles para el evento.	Lista de recursos.	Correcto
	PF_DP_15	Asignar recursos a actividades.	Listado de recursos disponibles para la actividad.	Lista de recursos.	Correcto
	PF_DP_16	Asignar recursos a compuertas.	Listado de recursos disponibles para la compuerta.	Lista de recursos.	Correcto
	PF_DP_17	Guardar recursos.	Notificación si la asignación se realizó sin problemas.	Notificación.	Correcto
EJECUCIÓN	PF_EP_01	Ver Procesos disponibles.	Obtiene lista de procesos a los que puede instanciar.	Listado de proceso.	Correcto
	PF_EP_02	Iniciar instancia de proceso.	Inicia la ejecución de un proceso.	Notificación de inicio de la instancia.	Correcto
	PF_EP_03	Revisar	Listado de	Lista de	Correcto

	notificación.	notificaciones.	notificaciones.	
<b>PF_EP_04</b>	Ver actividad.	Obtiene datos relacionados con la actividad.	Datos de actividad.	Correcto
<b>PF_EP_05</b>	Ejecutar actividad simple o de usuario.	Se guarda la actividad como realizada y se continúa con el flujo.	Notificación.	Correcto
<b>PF_EP_06</b>	Actividad automática.	Notifica a los involucrados y se continúa con el flujo.	Notificación	Correcto
<b>PF_EP_07</b>	Ejecutar actividad manual.	Se guarda la actividad como realizada y se continúa con el flujo.	Notificación	Correcto
<b>PF_EP_08</b>	Fin de instancia.	Notificación al supervisor.	Notificación de fin de Instancia.	Correcto

Elaboración. Autores

A continuación, en el Cuadro 85 se muestra en detalle las pruebas funcionales ejecutadas como supervisión del proceso para las etapas de diseño y ejecución, además de probar la generación de reportes generales del sistema.

**Cuadro 85.** Caso de prueba funcional en la supervisión de procesos.

ETAPA	CÓDIGO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
<b>DISEÑO</b>	<b>PF_SDP_01</b>	Ver proceso con lista de elementos.	Proceso con todos sus elementos.	Proceso y elementos.	Correcto
	<b>PF_SDP_02</b>	Modificar fecha límite del proceso.	Notificación del cambio de fecha.	Notificación.	Correcto
	<b>PF_SDP_03</b>	Modificar fecha límite de tarea.	Notificación del cambio de fecha.	Notificación.	Correcto
	<b>PF_SDP_04</b>	Modificar responsable de tarea.	Notificación del cambio de responsable.	Notificación.	Correcto
<b>EJECUCIÓN</b>	<b>PF_SEP_01</b>	Ver Procesos disponibles.	Obtiene lista de procesos para supervisar.	Lista de procesos.	Correcto
	<b>PF_SEP_02</b>	Ver detalles de proceso.	Obtiene la lista de instancias del proceso.	Lista de instancias del proceso.	Correcto
	<b>PF_SEP_03</b>	Ver detalles de instancia.	Obtiene los datos de la instancia tales como: trámites, archivos, errores, etc.	Elementos de la instancia.	Correcto
	<b>PF_SEP_04</b>	Reiniciar Instancia	Notificación de Reinicio.	Notificación.	Correcto

	<b>PF_SEP_05</b>	Eliminar Instancia	Notificación de eliminación.	Notificación.	Correcto
	<b>PF_SEP_06</b>	Reiniciar Actividad.	Notificación de reinicio.	Notificación.	Correcto
	<b>PF_GR_01</b>	Reporte ISO 9001.	Datos del proceso.	Datos del proceso	Correcto
<b>REPORTES</b>	<b>PF_GR_02</b>	Control de acceso.	Datos de acceso de usuarios especificados.	Datos de acceso de usuarios	Correcto
	<b>PF_GR_03</b>	Control de operadores.	Datos relacionados del operador, para verificar rendimiento.	Datos de rendimiento del operador.	Correcto

**Elaboración.** Autores

## ANEXO 7 – MANUAL DE PROGRAMADOR



**ESPAM MFL**  
ESCUELA SUPERIOR POLITÉCNICA  
AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ

# API WEB

## Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL



### MANUAL DEL PROGRAMADOR

Dayana H. Bailón Delgado  
José S. Mera Cantos

2017

## FICHA DEL DOCUMENTO

El cuadro 1 detalla las versiones que ha tenido este anexo, indicando: fecha, versión, autores y por quiénes fue revisado.

**Cuadro 86.** Ficha del documento.

Fecha	Versión	Autores	Verificado por
10/07/2017	1.0	Bailón Delgado Dayana Helen Mera Cantos José Simón	Ing. Joffre Moreira.

**Elaboración.** Autores

El cuadro 2 presenta las entidades que receptan y validan el contenido de este documento.

Documento validado por las partes en fecha:

**Cuadro 87.** Entidades que validan el documento.

Por el departamento	Por la universidad
Unidad de Producción de Software (UPS)	Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López (ESPAM MFL)

**Elaboración.** Autores

## CONTENIDO GENERAL

FICHA DEL DOCUMENTO .....	236
CONTENIDO GENERAL.....	237
CONTENIDO DE CUADROS Y FIGURAS.....	242
1 INTRODUCCIÓN .....	248
2 IDENTIFICACIÓN DEL SISTEMA.....	248
2.1. ALCANCE.....	249
2.2. OBJETIVO DEL SISTEMA .....	249
2.3. NORMAS, POLÍTICAS Y PROCEDIMIENTOS .....	249
2.4. REQUERIMIENTOS.....	250
2.4.1. REQUERIMIENTOS DE HARDWARE.....	250
2.4.2. HERRAMIENTAS DE SOFTWARE .....	250
2.4.2.1. VISUAL C# .....	251
2.4.2.2. VISUAL STUDIO.NET .....	251
2.4.2.3. SQL SERVER 2008 R2 EXPRESS .....	251
2.4.2.4. ASP.NET WEBAPI .....	251
2.4.3. CONOCIMIENTOS PREVIOS .....	251
3 ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....	252
3.1 VISTA FUNCIONAL.....	252
3.1.1 DIAGRAMA DE ESTADOS DE LOS ELEMENTOS DE LA APLICACIÓN.....	252
3.2 VISTA LÓGICA (ARQUITECTURA) .....	254
3.2.1 DESARROLLO EN CAPAS .....	255
3.2.1.1. MODELOS.....	255
3.2.1.2. RECURSOS WEB (CONTROLADORES) .....	257
3.3 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.....	258
4 DISEÑO Y ANÁLISIS DE ALGORITMOS .....	259
4.1 ESTANDARIZACIÓN DE CÓDIGO .....	259
4.2 ENTIDADES .....	260
4.3 CATÁLOGOS (LÓGICA) .....	261
4.3.1 MÉTODOS CRUD.....	262
4.3.2 ACCESO.....	266
4.3.3 ADJUNTO .....	266

4.3.4	ATRIBUTO.....	267
4.3.5	CABECERA TAREA .....	267
4.3.6	CAMPO TAREA.....	268
4.3.7	CARACTERÍSTICA.....	269
4.3.8	CARGO.....	270
4.3.9	CONTROL CABECERA.....	270
4.3.10	CONTROL INTEGRACIÓN.....	270
4.3.11	COORDENADA .....	271
4.3.12	DETALLE HORARIO .....	272
4.3.13	DETALLE REGISTRO .....	272
4.3.14	DETALLE TAREA .....	273
4.3.15	DÍA LABORABLE.....	274
4.3.16	EJECUCIÓN SUB PROCESO .....	274
4.3.17	ELEMENTO .....	275
4.3.18	ELEMENTO ATRIBUTO .....	276
4.3.19	ERROR.....	276
4.3.20	FORMULARIO.....	277
4.3.21	FORMULARIOELEMENTO .....	277
4.3.22	HISTORIAL INSTANCIA .....	279
4.3.23	HISTORIALTAREA .....	279
4.3.24	HORARIO TRABAJO.....	280
4.3.25	INSTANCIA.....	280
4.3.26	INTEGRACIÓN EXTERNA .....	282
4.3.27	INTEGRACIÓN INTERNA .....	283
4.3.28	MENSAJE TAREA .....	284
4.3.29	MENSAJE TRÁMITE .....	285
4.3.30	OBJETO FLUJO .....	286
4.3.31	OBJETO FLUJO ATRIBUTO .....	287
4.3.32	PARÁMETRO INTEGRACIÓN EXTERNA.....	287
4.3.33	PARÁMETRO INTEGRACIÓN INTERNA.....	287
4.3.34	PROCESO .....	288
4.3.35	QUERY STRING.....	289
4.3.36	REFERENCIA FORMULARIO .....	290

4.3.37	REFERENCIA FORMULARIO ELEMENTO .....	290
4.3.38	REFERENCIA INTEGRACIÓN .....	291
4.3.39	REGISTRO .....	291
4.3.40	REGISTRO EXTERNO .....	292
4.3.41	REGLA.....	292
4.3.42	REPORTE.....	293
4.3.43	RUTA .....	293
4.3.44	TAREA .....	294
4.3.45	TIPO DE DATO.....	295
4.3.46	TRÁMITE .....	295
4.3.47	USUARIO.....	298
4.3.48	USUARIO CARACTERÍSTICA .....	299
4.3.49	VERSIÓN.....	299
4.3.50	VERSIÓN ELEMENTO .....	300
4.3.51	WEB SERVICE .....	302
4.4	RECURSOS WEB (CONTROLADORES) .....	303
4.4.1	ACCESO.....	304
4.4.2	ACTUALIZAR HTML.....	305
4.4.3	APROBAR TAREA.....	305
4.4.4	ASIGNAR RECURSOS.....	306
4.4.5	ATRIBUTO.....	306
4.4.6	CABECERA .....	306
4.4.7	CARACTERÍSTICA.....	307
4.4.8	CARACTERÍSTICA USUARIO.....	307
4.4.9	CARGO.....	308
4.4.10	CONTAR INSTANCIA.....	308
4.4.11	CONTROL CABECERA.....	309
4.4.12	DEFINIR COMPUERTA.....	309
4.4.13	DETALLE FORMULARIO .....	309
4.4.14	DETALLE TAREA .....	310
4.4.15	DOCUMENTACIÓN 9001 .....	310
4.4.16	DOCUMENTOS INSTANCIA .....	311
4.4.17	EJECUCIÓN ATM.....	311

4.4.18	EJECUCIÓN FORMULARIO .....	311
4.4.19	EJECUCIÓN VDF COMPUERTA .....	312
4.4.20	EJECUCIÓN VDF .....	312
4.4.21	EJECUCIÓN VDI COMPUERTA.....	312
4.4.22	EJECUCIÓN VDI .....	313
4.4.23	ELEMENTO .....	313
4.4.24	FILTRO ADJUNTO .....	313
4.4.25	FITLRO AREA CABECERA.....	314
4.4.26	FILTRO AREA SUPERVISOR .....	314
4.4.27	FILRO AREA TAREA.....	315
4.4.28	FILTRO CABECERA.....	315
4.4.29	FILTRO CARGO .....	315
4.4.30	FILTRO FORMULARIO .....	316
4.4.31	FILTRO INSTANCIA .....	316
4.4.32	FILTRO INTEGRACIÓN .....	316
4.4.33	FILTRO INTEGRACIÓN FORMULARIO.....	317
4.4.34	FILTRO MENSAJE .....	317
4.4.35	FILTRO MENSAJE TRÁMITE.....	317
4.4.36	FILTRO PROCESO .....	318
4.4.37	FILTRO PROCESO DETALLE .....	318
4.4.38	FILTRO PROCESO ID.....	319
4.4.39	FILTRO REDISENHO .....	319
4.4.40	FILTRO REFERENCIA VD .....	319
4.4.41	FILTRO TAREA .....	320
4.4.42	FILTRO ÚLTIMAS ALERTAS .....	320
4.4.43	HISTORIAL INSTANCIA .....	321
4.4.44	HISTORIAL TAREA .....	321
4.4.45	HORARIO .....	321
4.4.46	INSTANCIA.....	322
4.4.47	INTEGRACIÓN EXTERNA .....	322
4.4.48	INTEGRACIÓN INTERNA .....	323
4.4.49	MENSAJE .....	323
4.4.50	MENSAJE TRÁMITE .....	324

4.4.51	OPERADOR PROCESO.....	324
4.4.52	PROCESO .....	325
4.4.53	REFERENCIA FORMULARIO .....	325
4.4.54	REFERENCIA INTEGRACIÓN .....	326
4.4.55	REGLA.....	326
4.4.56	REPORTE.....	326
4.4.57	RESPONSABLE .....	327
4.4.58	ROLES.....	327
4.4.59	SUPERVISIÓN .....	328
4.4.60	TAREA.....	328
4.4.61	TIPO DATO.....	328
4.4.62	TRÁMITE .....	329
4.4.63	USUARIO.....	329
4.4.64	VERSIÓN.....	330
4.4.65	VERSIÓN ELEMENTO .....	330
4.4.66	WEB SERVICE .....	330
5	METADATA.....	331
5.1	DATABASE FIRST .....	331
5.2	ESQUEMAS .....	331
5.3	DIAGRAMAS ENTIDAD-RELACIÓN.....	332
5.4	DESCRIPCIÓN DE TABLAS Y DATOS .....	334
5.5	PROCEDIMIENTOS ALMACENADOS.....	346
5.6	ROLES Y USUARIOS .....	351
6	GLOSARIO Y ABREVIATURAS .....	351
6.1	GLOSARIO.....	351
6.2	ABREVIATURAS.....	352
7	BIBLIOGRAFÍA .....	353

## CONTENIDO DE CUADROS Y FIGURAS

<b>Cuadro 1.</b> Ficha del documento.....	236
<b>Cuadro 2.</b> Entidades que validan el documento. ....	236
<b>Cuadro 3.</b> Identificación del sistema.....	248
<b>Cuadro 4.</b> Resumen de las herramientas de software.....	250
<b>Cuadro 5.</b> Características del Administrador.....	258
<b>Cuadro 6.</b> Características del Supervisor.....	259
<b>Cuadro 7.</b> Características del Operador.....	259
<b>Cuadro 8.</b> Características del Modelador de diagramas.....	259
<b>Cuadro 9.</b> Características del Diseñador.....	259
<b>Cuadro 10.</b> Características del Administrador de Base de Datos (DBA). ....	259
<b>Cuadro 11.</b> Nomenclatura.....	260
<b>Cuadro 12.</b> Métodos de LogicAcceso.....	266
<b>Cuadro 13.</b> Métodos de LogicAdjunto.....	266
<b>Cuadro 14.</b> Métodos de LogicAtributo.....	267
<b>Cuadro 15.</b> Métodos de LogicCabeceraTarea.....	267
<b>Cuadro 16.</b> Métodos de LogicCampoTarea.....	269
<b>Cuadro 17.</b> Métodos del LogicCaracterística.....	269
<b>Cuadro 18.</b> Métodos del LogicCargo.....	270
<b>Cuadro 19.</b> Métodos de LogicControlCabera.....	270
<b>Cuadro 20.</b> Métodos de LogicControllIntegración.....	271
<b>Cuadro 21.</b> Métodos de LogicCoordenada.....	271
<b>Cuadro 22.</b> Métodos del LogicDetalleHorario.....	272
<b>Cuadro 23.</b> Métodos de LogicControllIntegración.....	272
<b>Cuadro 24.</b> Métodos del LogicDetalleTarea.....	273
<b>Cuadro 25.</b> Métodos del LogicDiaLaborable.....	274
<b>Cuadro 26.</b> Métodos del LogicEjecuciónSubProceso.....	274
<b>Cuadro 27.</b> Métodos del LogicElemento.....	275
<b>Cuadro 28.</b> Métodos del LogicElementoAtributo.....	276
<b>Cuadro 29.</b> Métodos del LogicError.....	277
<b>Cuadro 30.</b> Métodos del LogicFormulario.....	277
<b>Cuadro 31.</b> Métodos del LogicFromularioElemento.....	278
<b>Cuadro 32.</b> Métodos del LogicHistorialInstancia.....	279

<b>Cuadro 33.</b> Métodos del LogicHistorialTarea.....	280
<b>Cuadro 34.</b> Métodos del LogicHorarioTrabajo .....	280
<b>Cuadro 35.</b> Métodos del LogicInstancia. ....	281
<b>Cuadro 36.</b> Métodos del LogicIntegraciónExterna.....	283
<b>Cuadro 37.</b> Métodos del LogicIntegraciónInterna. ....	283
<b>Cuadro 38.</b> Métodos del LogicMensajeTarea .....	284
<b>Cuadro 39.</b> Métodos del LogicMensajeTrámite .....	285
<b>Cuadro 40.</b> Métodos del LogicObjetoFlujo .....	286
<b>Cuadro 41.</b> Métodos del LogicObjetoFlujoAtributo .....	287
<b>Cuadro 42.</b> Métodos del LogicParámetroIntegraciónExterna .....	287
<b>Cuadro 43.</b> Métodos del LogicParámetroIntegraciónInterna .....	288
<b>Cuadro 44.</b> Métodos del LogicProceso.....	288
<b>Cuadro 45.</b> Métodos del LogicQueryString.....	289
<b>Cuadro 46.</b> Métodos del LogicReferenciaFormulario. ....	290
<b>Cuadro 47.</b> Métodos del LogicReferenciaFormularioElemento. ....	291
<b>Cuadro 48.</b> Métodos del LogicReferenciaIntegración.....	291
<b>Cuadro 49.</b> Métodos del LogicRegistro. ....	291
<b>Cuadro 50.</b> Métodos del LogicRegistroExterno. ....	292
<b>Cuadro 51.</b> Métodos del LogicRegla. ....	292
<b>Cuadro 52.</b> Métodos del LogicReporte. ....	293
<b>Cuadro 53.</b> Métodos del LogicRuta. ....	294
<b>Cuadro 54.</b> Métodos del LogicTarea. ....	294
<b>Cuadro 55.</b> Métodos del LogicTipoDato. ....	295
<b>Cuadro 56.</b> Métodos del LogicTrámite.....	296
<b>Cuadro 57.</b> Métodos del LogicUsuario. ....	298
<b>Cuadro 58.</b> Métodos del LogicUsuarioCaracteristica.....	299
<b>Cuadro 59.</b> Métodos del LogicVersión.....	299
<b>Cuadro 60.</b> Métodos del LogicVersiónElemento.....	300
<b>Cuadro 61.</b> Métodos del WebService. ....	302
<b>Cuadro 62.</b> Métodos de AccesoController. ....	304
<b>Cuadro 63.</b> Métodos HTTP de ActualizarHtmlController. ....	305
<b>Cuadro 64.</b> Métodos HTTP de AprobarTareaController. ....	306
<b>Cuadro 65.</b> Métodos HTTP de AsignarRecursosController. ....	306

<b>Cuadro 66.</b> Métodos HTTP de AtributoController. ....	306
<b>Cuadro 67.</b> Métodos HTTP de CabeceraController. ....	307
<b>Cuadro 68.</b> Métodos HTTP de CaracterísticaController. ....	307
<b>Cuadro 69.</b> Métodos HTTP de CaracterísticaUsuarioController. ....	308
<b>Cuadro 70.</b> Métodos HTTP de CargoController. ....	308
<b>Cuadro 71.</b> Métodos HTTP de ContarInstanciaController. ....	308
<b>Cuadro 72.</b> Métodos HTTP de ControlCabeceraController. ....	309
<b>Cuadro 73.</b> Métodos HTTP de DefinirCompuertaController. ....	309
<b>Cuadro 74.</b> Métodos HTTP de DetalleFormularioController. ....	310
<b>Cuadro 75.</b> Métodos HTTP de DetalleTareaController. ....	310
<b>Cuadro 76.</b> Métodos HTTP de Documentación9001Controller. ....	311
<b>Cuadro 77.</b> Métodos HTTP de DocumentosInstanciaController. ....	311
<b>Cuadro 78.</b> Métodos HTTP de EjecuciónATMController. ....	311
<b>Cuadro 79.</b> Métodos HTTP de EjecuciónFormularioController. ....	312
<b>Cuadro 80.</b> Métodos HTTP de EjecuciónVDFCompuertaController. ....	312
<b>Cuadro 81.</b> Métodos HTTP de EjecuciónVDFController. ....	312
<b>Cuadro 82.</b> Métodos HTTP de EjecuciónVDICompuertaController. ....	313
<b>Cuadro 83.</b> Métodos HTTP de EjecuciónVDIController. ....	313
<b>Cuadro 84.</b> Métodos HTTP de ElementoController. ....	313
<b>Cuadro 85.</b> Métodos HTTP de FiltroAdjuntoController. ....	314
<b>Cuadro 86.</b> Métodos HTTP de FiltroAreaCabeceraController. ....	314
<b>Cuadro 87.</b> Métodos HTTP de FiltroAreaSupervisorController. ....	314
<b>Cuadro 88.</b> Métodos HTTP de FiltroAreaTareaController. ....	315
<b>Cuadro 89.</b> Métodos HTTP de FiltroCabeceraController. ....	315
<b>Cuadro 90.</b> Métodos HTTP de FiltroCargoController. ....	316
<b>Cuadro 91.</b> Métodos HTTP de FiltroFormularioController. ....	316
<b>Cuadro 92.</b> Métodos HTTP de FiltroInstanciaController. ....	316
<b>Cuadro 93.</b> Métodos HTTP de FiltroIntegraciónController. ....	317
<b>Cuadro 94.</b> Métodos HTTP de FiltroIntegraciónFormularioController. ....	317
<b>Cuadro 95.</b> Métodos HTTP de FiltroMensajeController. ....	317
<b>Cuadro 96.</b> Métodos HTTP de FiltroMensajeTramiteController. ....	318
<b>Cuadro 97.</b> Métodos HTTP de FiltroProcesoController. ....	318
<b>Cuadro 98.</b> Métodos HTTP de FiltroProcesoDetalleController. ....	319

<b>Cuadro 99.</b> Métodos HTTP de FiltroProcesoldController. ....	319
<b>Cuadro 100.</b> Métodos HTTP de FiltroRedisenhoController. ....	319
<b>Cuadro 101.</b> Métodos HTTP de FiltroReferenciaVDController. ....	320
<b>Cuadro 102.</b> Métodos HTTP de FiltroTareaController. ....	320
<b>Cuadro 103.</b> Métodos HTTP de FiltroÚltimasAlertasController. ....	320
<b>Cuadro 104.</b> Métodos HTTP de HistorialInstanciaController. ....	321
<b>Cuadro 105.</b> Métodos HTTP de HistorialTareaController. ....	321
<b>Cuadro 106.</b> Métodos HTTP de HorarioController.....	322
<b>Cuadro 107.</b> Métodos HTTP de InstanciaController. ....	322
<b>Cuadro 108.</b> Métodos HTTP de IntegraciónExternaController. ....	323
<b>Cuadro 109.</b> Métodos HTTP de IntegraciónInternaController.....	323
<b>Cuadro 110.</b> Métodos HTTP de MensajeController.....	324
<b>Cuadro 111.</b> Métodos HTTP de MensajeTramiteController.....	324
<b>Cuadro 112.</b> Métodos HTTP de FiltroMensajeController.....	325
<b>Cuadro 113.</b> Métodos HTTP de ProcesoController. ....	325
<b>Cuadro 114.</b> Métodos HTTP de ReferenciaFormularioController.....	325
<b>Cuadro 115.</b> Métodos HTTP de ReferencialIntegraciónController. ....	326
<b>Cuadro 116.</b> Métodos HTTP de ReglaController.....	326
<b>Cuadro 117.</b> Métodos HTTP de ReporteController.....	327
<b>Cuadro 118.</b> Métodos HTTP de ResponsableController. ....	327
<b>Cuadro 119.</b> Métodos HTTP de RolesController. ....	327
<b>Cuadro 120.</b> Métodos HTTP de SupervisiónController.....	328
<b>Cuadro 121.</b> Métodos HTTP de TareaController.....	328
<b>Cuadro 122.</b> Métodos HTTP de TipoDatoController.....	329
<b>Cuadro 123.</b> Métodos HTTP de TrámiteController. ....	329
<b>Cuadro 124.</b> Métodos HTTP de UsuarioController. ....	329
<b>Cuadro 125.</b> Métodos HTTP de VersiónController. ....	330
<b>Cuadro 126.</b> Métodos HTTP de VersiónElementoController. ....	330
<b>Cuadro 127.</b> Métodos HTTP de WebServiceController. ....	331
<b>Cuadro 128.</b> Descripción de los esquemas de la BD. ....	331
<b>Cuadro 129.</b> Tablas pertenecientes a la BD Motor_Workflow .....	334
<b>Cuadro 130.</b> Columnas correspondientes a las tablas del esquema general (.dbo).....	335

<b>Cuadro 131.</b> Columnas correspondientes a las tablas del esquema "diseñoproceso".	337
<b>Cuadro 132.</b> Columnas correspondientes a las tablas del esquema "instanciaproceso".	339
<b>Cuadro 133.</b> Columnas correspondientes a las tablas del esquema "tarea".	341
<b>Cuadro 134.</b> Columnas correspondientes a las tablas del esquema "usuario".	346
<b>Cuadro 135.</b> Credenciales de acceso a la BD Motor_Workflow.	351
<b>Figura 1.</b> Estados y transición durante la ejecución de un proceso administrativo.	253
<b>Figura 2.</b> Estados y transición de las tareas o trámites durante la ejecución de un proceso.	253
<b>Figura 3.</b> Estados y transición de la comunicación (notificaciones y alertas) entre los usuarios, durante la ejecución del proceso.	253
<b>Figura 4.</b> Estados y transición de la ejecución de las tareas durante la creación o diseño del proceso.	254
<b>Figura 5.</b> Estados y transición de los mensajes enviados a los usuarios, durante el diseño del proceso.	254
<b>Figura 6.</b> Estructura de la Solución "motorWorkflow".	255
<b>Figura 7.</b> Entidades de la solución motorWorkflow.	256
<b>Figura 8.</b> Catálogos de la solución motorWorkflow.	257
<b>Figura 9.</b> Controladores de la solución motorWorkflow.	258
<b>Figura 10.</b> Tabla detallada de la entidad ObjetoFlujo.	261
<b>Figura 11.</b> Entidad correspondiente a la tabla ObjetoFlujo.	261
<b>Figura 12.</b> Objetos correspondientes a la instancia del WorkflowEntities y del LogicQueryStrig.	262
<b>Figura 13.</b> Relaciones entre las tablas HorarioTrabajo, DetalleHorario, DiasLaborables.	263
<b>Figura 14.</b> Método GuardarHorario de LogicHorario.	263
<b>Figura 15.</b> Método ObtenerHorario de LogicHorario.	264
<b>Figura 16.</b> Método ActualizarHorario de LogicHorario.	265
<b>Figura 17.</b> Método EliminarHorario de LogicHorario.	265

<b>Figura 18.</b> Ejemplo de herencia de ApiController. ....	303
<b>Figura 19.</b> Código del método GET de AccesoController. ....	304
<b>Figura 20.</b> Código del método POST de AccesoController. ....	305
<b>Figura 21.</b> Diagrama del esquema usuario.....	332
<b>Figura 22.</b> Diagrama del esquema instanciaproceso.....	332
<b>Figura 23.</b> Diagrama del esquema tarea. ....	333
<b>Figura 24.</b> Diagrama del esquema diseñoproceso. ....	333
<b>Figura 25.</b> ConnectionStrings. ....	334
<b>Figura 26.</b> Procedimientos almacenados que no pertenecen a ningún esquema. ....	347
<b>Figura 27.</b> Procedimientos almacenados pertenecientes al esquema "diseñoproceso". ....	348
<b>Figura 28.</b> Procedimientos almacenados pertenecientes al esquema "instanciaproceso".....	349
<b>Figura 29.</b> Procedimientos almacenados pertenecientes al esquema "tarea". .....	350
<b>Figura 30.</b> Procedimientos almacenados pertenecientes al esquema "usuario". .....	351

## INTRODUCCIÓN

Este documento es un Manual del programador para el núcleo de un Workflow de procesos administrativos en la ESPAM MFL. Aunque MEIDAW (metodología utilizada durante el desarrollo) no especifica la realización de este documento, es un requerimiento determinado por la entidad encargada de la validación del sistema, por lo cual se ha estructurado basándose en las directrices brindadas por la UPS (Unidad de Producción de Software).

Este documento está dirigido a los encargados de dar el mantenimiento necesario del núcleo del Workflow, mismo que permitirá dar a conocer todas las funcionalidades del núcleo, mediante la descripción de las entidades, métodos, variables y demás elementos de la plataforma.

## IDENTIFICACIÓN DEL SISTEMA

A continuación, en el Cuadro 88 se detalla información principal de identificación del Núcleo de Workflow desarrollado.

**Cuadro 88.** Identificación del sistema.

Parámetros	Detalles
Nombre del Sistema	Núcleo del Workflow
Versión	1.0
Logotipo	
Área de desarrollo	Desarrollado en la Carrera de Computación para todas las áreas donde se realicen procesos administrativos de la ESPAM MFL
ERS de referencia	Anexo 2.
Desarrollador / Equipo	Dayana Bailón, José Mera.
Modelo de desarrollo	Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow.
Paradigma de programación	Programación orientada a objetos.

Elaboración. Autores

## **2.1. ALCANCE**

El Manual del programador detalla de forma precisa los métodos y procedimientos de cada clase o entidad encargada de las diferentes acciones o eventos realizados por el “Núcleo de un Workflow de Procesos Administrativos de la ESPAM MFL”. Además, se incluye el diagrama entidad-relación de la Base de datos (BD) del sistema y su respectivo diccionario de datos.

La versión 1.0 de este documento se ha realizado con base a la codificación o programación del sistema durante la ejecución de los incrementos planificatorios detallados de forma independiente en el Anexo 5, mismos que tuvieron como fin alcanzar lo especificado en el Plan de proyecto (Anexo 1), mediante investigación bibliográfica por parte de los autores y a las correcciones realizadas por el tutor. El avance de la codificación del sistema y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento, produciendo nuevas versiones actualizadas.

## **2.2. OBJETIVO DEL SISTEMA**

El objetivo del núcleo es procesar la información de las actividades que intervienen en los diferentes procesos administrativos de la ESPAM MFL para asignarlas a los recursos correspondientes, ya sean humanos o tecnológicos, para que cumplan con las respectivas tareas: manuales o automáticas.

## **2.3. NORMAS, POLÍTICAS Y PROCEDIMIENTOS**

A fin de facilitar la creación, desarrollo y análisis del núcleo del Workflow, se adaptó el estándar establecido por la Workflow Management Coalition (WFMC): WfMC-TC-1003 v1.1. Por otra parte, para el diseño de los diagramas de proceso que gestiona el núcleo del Workflow se siguió el estándar BPMN 2.0 (Notación de Modelado de Procesos de Negocio).

En cuanto a las restricciones de las herramientas de programación utilizadas, se adaptaron las recomendaciones brindadas por la UPS, por ejemplo utilizar

SQL Server como Sistema Gestor de Base de Datos (SGBD), a fin de adaptar el sistema a su ambiente de producción.

## 2.4. REQUERIMIENTOS

### 2.4.1. REQUERIMIENTOS DE HARDWARE

Los servidores donde se implemente la API web, deben ser capaces de atender consultas concurrentemente. El sistema ha sido diseñado mediante un modelo cliente/servidor, siendo el núcleo el servidor, preparado para atender las peticiones de las aplicaciones clientes (interfaces). Los recursos de la API web deberán ser consumidos preferiblemente por las interfaces desarrolladas por las autoras de la tesis complementaria.

Las características principales del servidor son:

- Servidor Intel Xeon.
- 2.53 GHz.
- RAM de 6 GB
- 15 MB de cache, 64 bits
- Disco duro de 2 teras 7200 rpm SATA3.
- Windows Server 2008 R2.

### 2.4.2. HERRAMIENTAS DE SOFTWARE

En el Cuadro 89 se presenta un breve resumen de las herramientas utilizadas y a continuación la respectiva descripción. El análisis de las tecnologías disponibles en la actualidad para desarrollar el núcleo de un sistema Workflow está especificado en el Anexo 1, el Plan de Proyecto.

**Cuadro 89.** Resumen de las herramientas de software.

Lenguaje de programación	Entorno de Desarrollo Integrado (IDE)	Sistema Gestor de Base de Datos (SGBD)	Framework
Visual C#	Visual Studio (VS)	SQL Server 2008 R2 Express	ASP.NET WebAPI

Elaboración: Autores

#### **2.4.2.1. VISUAL C#**

Stein & Geyer-Schulz (2013) indican que es un lenguaje de programación que se destaca por su robustez y sencillez, basado en el paradigma POO (Programación Orientada a Objetos) y que combina el tipado de sistema estático junto con el dinámico, utilizando como estrategia de ejecución JIT-Compiled, mediante su CLR, destinado para compilar CIL al código de maquina nativo.

#### **2.4.2.2. VISUAL STUDIO.NET**

De acuerdo a Arbeláez *et al.* (2011) soporta varios lenguajes de programación como lo son Visual C#, ASP.NET, entre otros. En este aspecto, Microsoft (2011) citado por Arbeláez *et al.* (2011) detalla que Visual Studio es un Entorno de Desarrollo Integrado (IDE) que permite a los desarrolladores de software crear aplicaciones de escritorio, aplicaciones de tipo cliente servidor, aplicaciones y servicios web, entre otras.

#### **2.4.2.3. SQL SERVER 2008 R2 EXPRESS**

Martín *et al.* (2016) especifican que es un Sistema Gestor de Base de Datos (SGBD) basado en el lenguaje Transact-SQL, que según Fotache & Strimbei (2015) es capaz de analizar grandes conjuntos de tablas relacionales y el procesamiento de modelos estadísticos. Express se refiere a la versión liviana y gratuita.

#### **2.4.2.4. ASP.NET WEBAPI**

De acuerdo al sitio oficial de Microsoft (2015), ASP.NET Web API es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles. ASP.NET Web API es la plataforma perfecta para crear aplicaciones RESTful en .NET Framework.

### **2.4.3. CONOCIMIENTOS PREVIOS**

- Visual C#.

- Recursos web, peticiones HTTP.
- ASP.Net Web API.
- Procedimientos almacenados, tablas relacionadas, esquemas, y otras funciones de SQL Server.
- Estándar BPMN 2.0.
- Lectura y escritura de archivos XML y HTML.

## **ANÁLISIS Y DISEÑO DE LA APLICACIÓN**

### **1.8 VISTA FUNCIONAL**

#### **1.8.1 DIAGRAMA DE ESTADOS DE LOS ELEMENTOS DE LA APLICACIÓN**

En la Figura 28 se detallan los estados que puede tener un proceso administrativo durante su ejecución o instancia, mismo que contiene trámites (actividades) realizadas por los operadores o usuarios del proceso, por lo tanto, en la Figura 29 se presentan las posibles fases de dichos trámites. Cabe recalcar, que se establece como sistema de comunicación el envío y recepción de notificaciones y alertas entre los participantes durante la ejecución de procesos (Figura 30).

Antes de que un proceso pueda ser ejecutado, se debe llevar a cabo una serie de tareas para su diseño o preparación, incluyendo la asignación de recursos (tiempo, usuario responsable, formulario, etc.), para lo cual se presenta en la Figura 31 los estados y transición de las tareas realizadas por los usuarios para la creación de los procesos administrativos. Finalmente, en la Figura 32 se detalla la forma de comunicación entre el sistema y los responsables de dichas tareas.

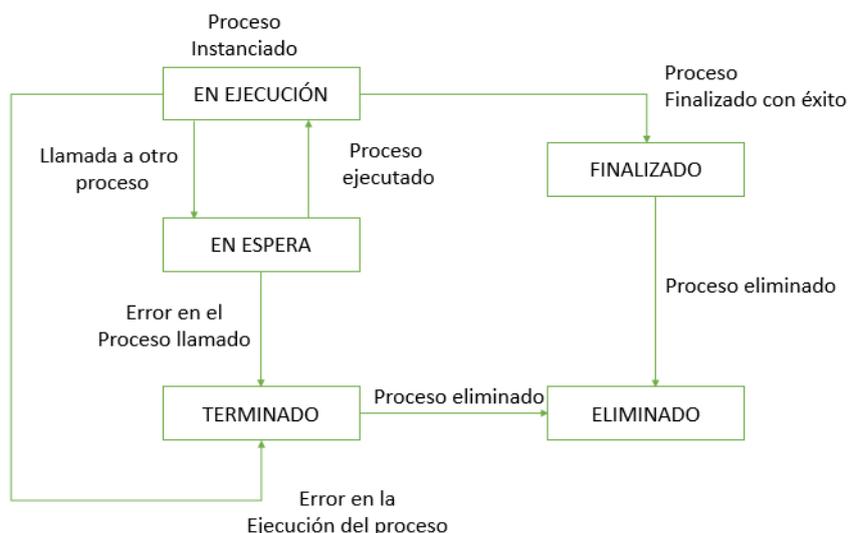


Figura 28. Estados y transición durante la ejecución de un proceso administrativo.

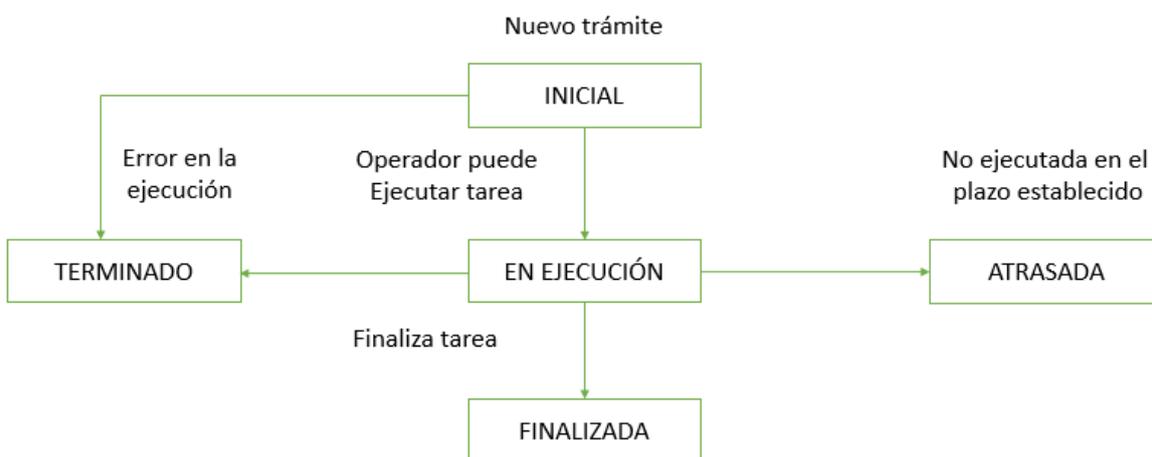


Figura 29. Estados y transición de las tareas o trámites durante la ejecución de un proceso.

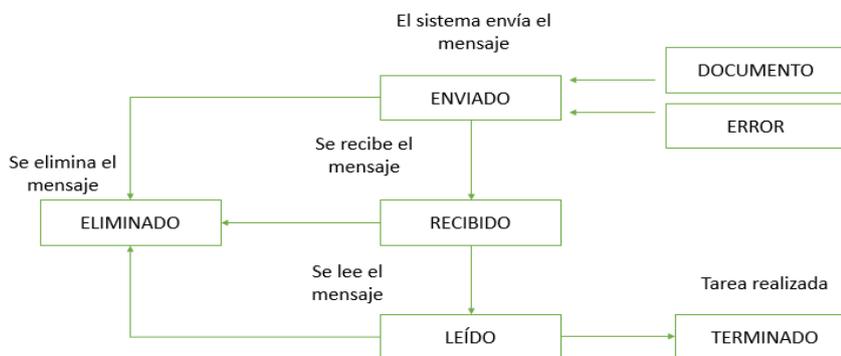


Figura 30. Estados y transición de la comunicación (notificaciones y alertas) entre los usuarios, durante la ejecución del proceso.

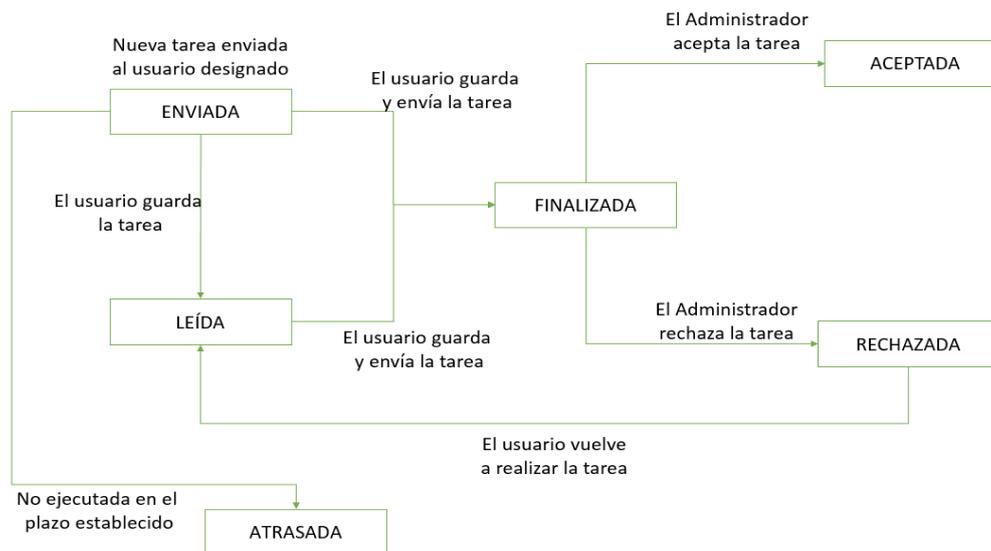


Figura 31. Estados y transición de la ejecución de las tareas durante la creación o diseño del proceso.

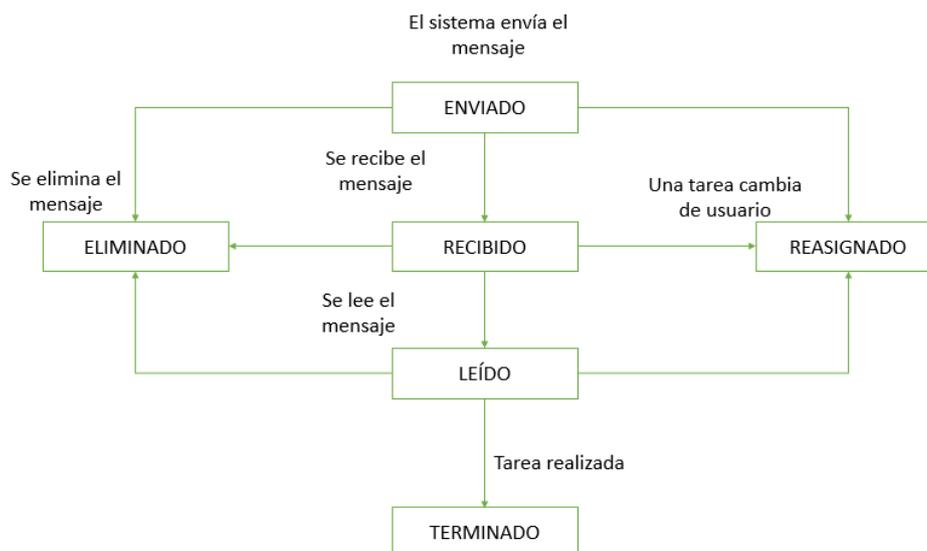


Figura 32. Estados y transición de los mensajes enviados a los usuarios, durante el diseño del proceso.

## 1.9 VISTA LÓGICA (ARQUITECTURA)

La solución tiene el nombre *motorWorkflow* y es un proyecto de Visual C#, del subconjunto web, del tipo ASP.NET Web Application. La plantilla utilizada es la de Web Api. Luego de que se crea el proyecto, se realiza la respectiva conexión y se agregan las carpetas: *Entidades* y *Logic* dentro del directorio *Models*, a fin de manejar las entidades o clases de la BD y sus respectivos

catálogos o métodos de la lógica del negocio, respectivamente. La estructura completa de la solución se detalla en la Figura 33.

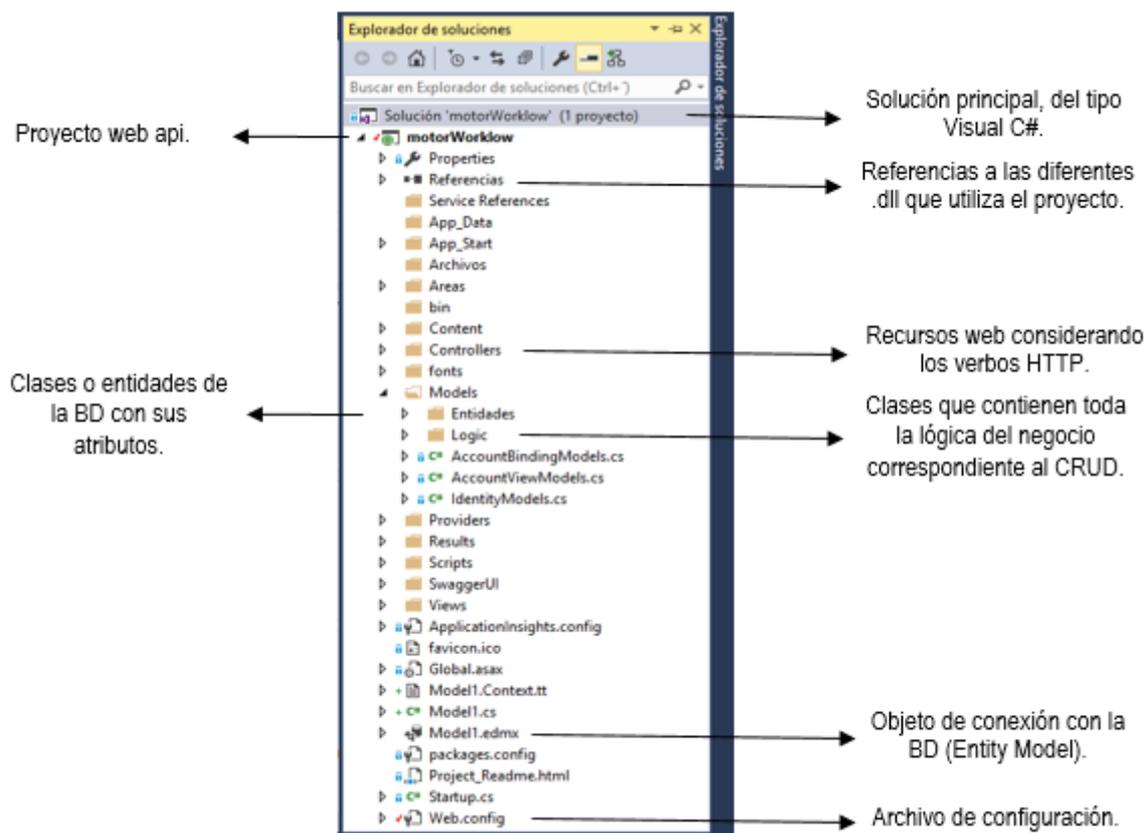


Figura 33. Estructura de la Solución "motorWorkflow".  
Elaboración: Autores.

## 1.9.1 DESARROLLO EN CAPAS

### MODELOS

Es una de las carpetas más importantes del proyecto, dado que contiene un directorio para las entidades (clases) correspondientes a las tablas de la BD, con sus respectivos atributos, y otro directorio para los catálogos que contienen los métodos que permiten la conexión directa con los procedimientos almacenados de la BD.

### ENTIDADES

Representan las tablas que se encuentran en la BD Motor\_Workflow, cada entidad presentada en la Figura 34 contiene un conjunto de parámetros, la

mayoría de ellos corresponden a los campos de las tablas y otros fueron incluidos únicamente como variables necesarias para la ejecución del sistema.

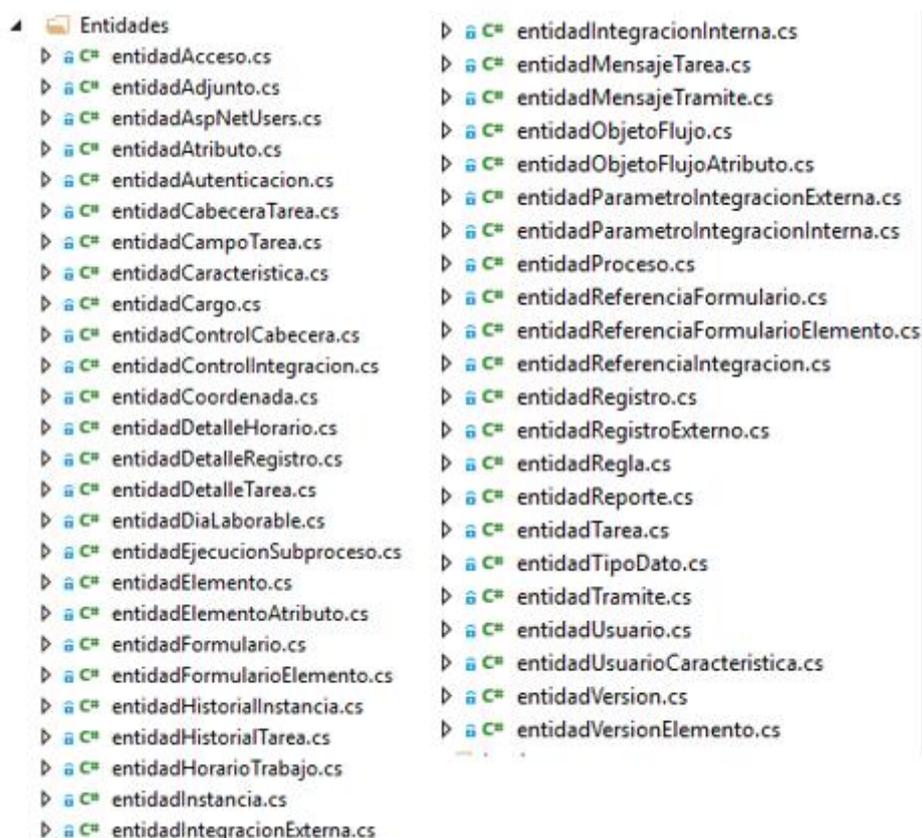
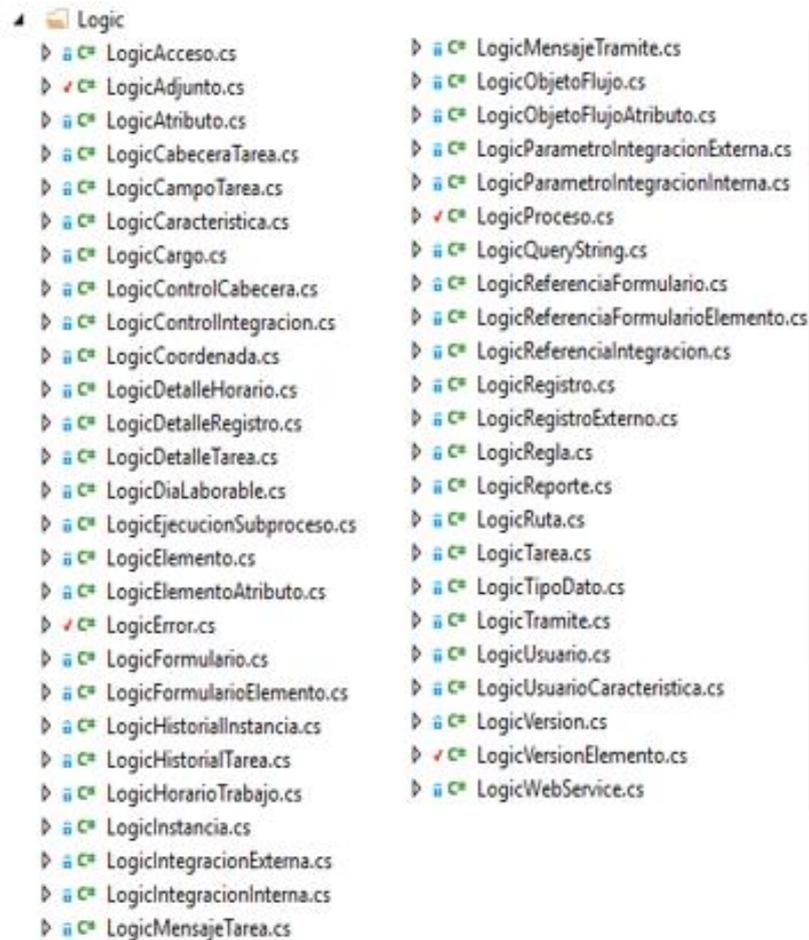


Figura 34. Entidades de la solución motorWorkflow.

Elaboración: Autores.

### CATÁLOGOS (LÓGICA)

Corresponden a la lógica del sistema, es la parte medular de núcleo del Workflow dado que se encarga del procesamiento de los datos consultados a la BD del sistema y posteriormente enviados mediante las peticiones HTTP a las aplicaciones clientes, o en su defecto realizar el proceso contrario; se recibe información mediante la URL de las peticiones, los métodos y procedimientos de los catálogos procesan y envían los datos a las respectivas tablas de la BD. Cada entidad presentada en la Figura 34 contiene un catalogo de la Figura 35 para la gestión de sus datos o información.



**Figura 35.** Catálogos de la solución motorWorkflow.  
Elaboración: Autores.

## RECURSOS WEB (CONTROLADORES)

Permiten enviar y recibir información por parte de las aplicaciones cliente. Los recursos gestionados por la API siguen el estilo arquitectónico REST. Los controladores del sistema están detallados en la Figura 36.

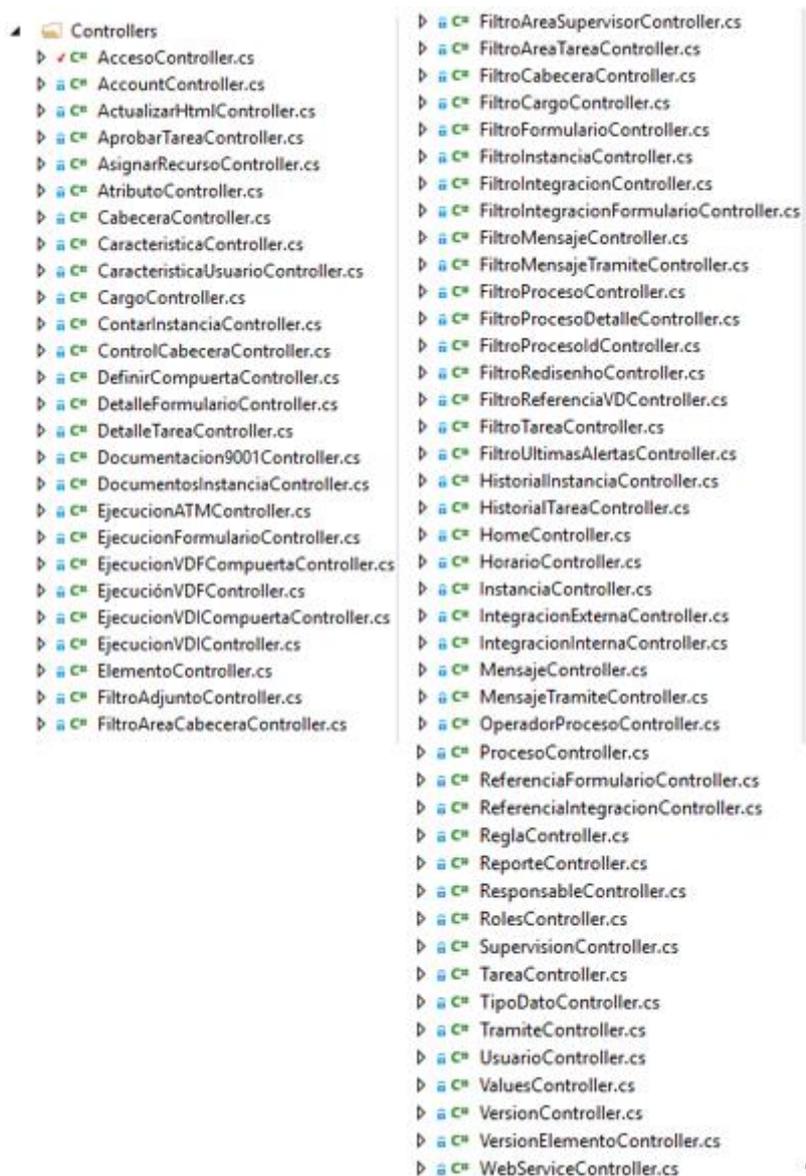


Figura 36. Controladores de la solución motorWorkflow.  
Elaboración: Autores.

## 1.10 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA

En esta parte se describen los roles de usuario y las actividades que pueden realizar dentro de la aplicación Workflow (Cuadros 5, 6, 7, 8, 9 y 10).

Cuadro 90. Características del Administrador.

Tipo de usuario	Administrador
Actividades	Puede aprobar o no un nuevo proceso, revisar el cumplimiento de todos los trámites que intervienen en el proceso que administra, reasignar responsables y realizar reportes de dichos procesos.

Elaboración: Autores

**Cuadro 91.** Características del Supervisor.

<b>Tipo de usuario</b>	<b>Supervisor</b>
<b>Actividades</b>	Puede verificar el cumplimiento de todos los trámites del proceso, reasignar responsables en las actividades y realizar reportes de los procesos.
<b>Elaboración:</b> Autores	

**Cuadro 92.** Características del Operador.

<b>Tipo de usuario</b>	<b>Operador</b>
<b>Actividades</b>	Realiza únicamente los trámites (tareas) a los que ha sido asignado. En caso de que en la etapa de diseño del proceso, durante la asignación de recursos, haya sido definido como usuario responsable del inicio del proceso, puede instanciarlo cuando sea necesario.
<b>Elaboración:</b> Autores	

**Cuadro 93.** Características del Modelador de diagramas.

<b>Tipo de usuario</b>	<b>Modelador de diagramas</b>
<b>Actividades</b>	Se encarga de modelar el diagrama (flujo) del proceso.
<b>Elaboración:</b> Autores	

**Cuadro 94.** Características del Diseñador.

<b>Tipo de usuario</b>	<b>Diseñador</b>
<b>Actividades</b>	Es el responsable de diseñar los formularios que posteriormente permitirán que el Operador registre la respectiva información, además puede generar Vistas de datos y Reportes en PDF que presenten información relevante de los formularios, añadiéndole diseño personalizado.
<b>Elaboración:</b> Autores	

**Cuadro 95.** Características del Administrador de Base de Datos (DBA).

<b>Tipo de usuario</b>	<b>DBA (Administrador de Base de Datos)</b>
<b>Actividades</b>	Es el encargado utilizar servicios web cuando se requieran integraciones, ya sea a servicios externos o a la información propia del Núcleo.
<b>Elaboración:</b> Autores	

## DISEÑO Y ANÁLISIS DE ALGORITMOS

### 1.11 ESTANDARIZACIÓN DE CÓDIGO

La nomenclatura utilizada se presenta de forma resumida en el Cuadro 96, establecido de acuerdo a las especificaciones del Manual de código fuente y estándares de desarrollo, elaborado y facilitado por la UPS.

Cuadro 96. Nomenclatura.

	Tipo general	Tipo específico	Nomenclatura	Ejemplo
Web api	Clases	Logic	UpperCamelCase	LogicProceso
		Entidades	lowerCamelCase	entidadProceso
	Variables	Globales	Todo en mayúsculas	PROCESO
		Locales	Se antepone el subguión (_) seguido del nombre que se le quiera dar a la variable	_id_proceso
		Objetos de catálogos (logic)	Se antepone el subguión (_) seguido de la abreviación de objeto "obj" y posteriormente el nombre.	_objCatalogoProceso
		Objetos de entidades		_objProceso
Listas	-	Se antepone el subguión (_) seguido de la palabra "lista" y posteriormente otro subguión (_) seguido del nombre.	_lista_procesos	
Métodos	-	UpperCamelCase y colocando el nombre de la acción principal que realiza el método.	GuardarProceso	
Base de datos	Base de datos	-	Snake case, se escribe cada primera letra de una nueva palabra con mayúscula, sin excederse de 3 palabras y sin utilizar las palabras "base" y "datos"	Motor_Workflow
	Esquemas	-	Todo en minúsculas.	instanciaproceso
	Tablas	-	UpperCamelCase y colocando el nombre de la tabla que corresponda, sin excederse de 2 palabras ni utilizar la palabra "tabla".	Proceso
	Campos de las tablas	-	En minúsculas salvo en el caso de que sean más de 1 palabra en el cual se utiliza el estilo de escritura lowerCamelCase.	idProceso
	Stores Procedures	-	Se antepone la palabra "Sp_" seguido de la acción que realizara este procedimiento y de la tabla a la cual afectara, todo esto utilizando el estilo de escritura UpperCamelCase.	Sp_GuardarProceso

Elaboración: Autores.

## 1.12 ENTIDADES

Como ya se mencionó, corresponde a las tablas que han sido previamente diseñadas en la BD y dado que en el capítulo 5 de este documento se incluye el diccionario de datos, solo se ubica como ejemplo la *entidadObjetoFlujo*. En la Figura 37 se muestra la tabla almacenada en la BD y en la Figura 38 se presenta la correspondiente clase registrada en el sistema.

ObjetoFlujo (diseñoproceso)			
	Column Name	Data Type	Allow Nulls
🔑	idObjetoFlujo	nchar(10)	<input type="checkbox"/>
	nombre	varchar(100)	<input type="checkbox"/>
	descripcion	varchar(MAX)	<input type="checkbox"/>
	categoria	nchar(2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 37. Tabla detallada de la entidad ObjetoFlujo.  
Elaboración: Autores.

```

/// <summary>
/// Contiene los objetos de flujo establecidos por el estándar BPMN 2.0.
/// Estos datos han sido previamente almacenados en la BD.
/// </summary>
14 referencias | Jose Simon Mera Cantos, Hace 29 días | 2 autores, 4 cambios
public class entidadObjetoFlujo
{
    /// <summary>
    /// Constructor vacío.
    /// </summary>
    5 referencias | Jose Simon Mera Cantos, Hace 29 días | 2 autores, 3 cambios
    public entidadObjetoFlujo()
    {
    }

    /// <summary>
    /// Clave primaria, no es autogenerada.
    /// </summary>
    2 referencias | Jose Simon Mera Cantos, Hace 29 días | 2 autores, 3 cambios
    public string _id_objeto_flujo { get; set; }
    /// <summary>
    /// Nombre del objeto, según el estándar BPMN 2.0.
    /// </summary>
    1 referencia | Jose Simon Mera Cantos, Hace 29 días | 2 autores, 3 cambios
    public string _nombre { get; set; }
    /// <summary>
    /// Descripción corta del objeto, según el estándar BPMN 2.0.
    /// </summary>
    2 referencias | Jose Simon Mera Cantos, Hace 29 días | 2 autores, 3 cambios
    public string _descripcion { get; set; }
    /// <summary>
    /// Tipo de objeto: Compuerta (C), evento (E) y Tarea (T)
    /// </summary>
    0 referencias | 0 cambios | 0 autores, 0 cambios
    public string _categoria { get; set; }
}

```

Figura 38. Entidad correspondiente a la tabla ObjetoFlujo.  
Elaboración: Autores.

### 1.13 CATÁLOGOS (LÓGICA)

Es necesario mencionar que existen dos objetos que están instanciados al inicio de todas las clases de la carpeta *Logic*. En la Figura 39 se presentan los siguientes objetos:

- ***\_objConexion***, que permite mencionar cada SP que ha sido añadido en el Entity, es decir, es el responsable de la conexión entre el sistema y la BD.

- **\_objEn**, que permite utilizar los métodos de encriptación y desencriptación, que han sido creados para mantener la integridad de los datos gestionados por el sistema, y son explicados más adelante.

```

/// <summary>
/// Para acceder a los procedimientos almacenados de la BD Motor_Workflow.
/// </summary>
WorkflowEntities _objConexion = new WorkflowEntities();

/// <summary>
/// Para acceder a los métodos de encriptación y desencriptación de la API.
/// </summary>
LogicQueryString _objEn = new LogicQueryString();

```

**Figura 39.** Objetos correspondientes a la instancia del WorkflowEntities y del LogicQueryStrig.

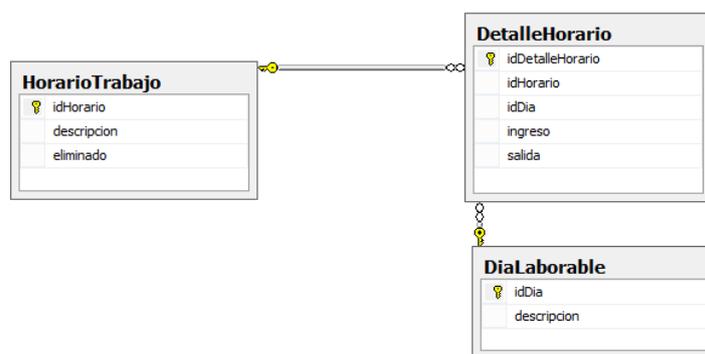
**Elaboración:** Autores.

### 1.13.1 MÉTODOS CRUD

En cuanto a los métodos de esta carpeta, hay que detallar que dado que en la BD *Motor\_Workflow* existen dos tipos de SP; en primer lugar, para las funciones básicas en todos los sistemas, también conocidos como CRUD, y los demás métodos, es decir, los que pueden (o no) utilizar algún método CRUD, pero que no solo realizan tal operación, sino que desencadenan un conjunto de comandos u órdenes para alcanzar otro tipo de acción específica, por ejemplo, calcular las horas disponibles para que un trabajador pueda realizar alguna tarea, según la fecha y hora límite, la fecha y hora de envío y el horario concreto de dicho trabajador (disponibilidad). Por lo cual, en la carpeta *Logic* del sistema se puede reconocer si un método pertenece al CRUD por el nombre, de acuerdo a la acción: guardar=crear, obtener=leer, actualizar y eliminar se mantienen iguales, mientras que los métodos con nombre diferente, serán ajenos al CRUD.

Dado que para la mayoría de entidades se realizaron las funciones CRUD, solo se utiliza la clase *LogicHorario* para explicar detalladamente su funcionamiento, permitiendo que el lector pueda deducir que los demás métodos tienen la misma lógica, según el tipo de acción. Se elige esta entidad porque mantiene una relación “de uno a muchos” (1-n) con la tabla *DetalleHorario* y a su vez

dicha tabla tiene con la tabla *DiaLaborable* una relación “uno a uno” (1-1), por lo cual se facilita el entendimiento para las tablas relacionadas. Estas tablas y sus relaciones no han sido presentadas en la sección de esquemas, porque son independientes (.dbo), por lo tanto se muestran en la Figura 40.



**Figura 40.** Relaciones entre las tablas HorarioTrabajo, DetalleHorario, DiasLaborables.  
**Elaboración:** Autores.

En la Figura 41, el método *GuardarHorario* recibe la instancia del *\_objHorario* que se va a enviar a la BD, posteriormente llama al *Sp\_GuardarHorario* y envía dicho objeto. Se captura el *\_id\_horario* que devuelve el SP, dado que se necesita guardar los detalles de ese horario y se recorre la *\_listaCrear*, que corresponde a los nuevos detalles del horario y cada ítem se envía a la BD mediante el *Sp\_GuardarDetalleHorario*, para lo cual se crea una nueva instancia de la *entidadDetalleHorario*, misma que se llena con los parámetros contenidos en cada ítem de la lista mencionada.

```

/// <summary>
/// Guarda un nuevo horario, incluido el detalle.
/// </summary>
/// <param name="_objHorarioTrabajo">objeto horario de trabajo</param>
1 referencia
public void GuardarHorario(entidadHorarioTrabajo _objHorarioTrabajo)
{
    int? _id_horario = _objConexion.Sp_GuardarHorario(_objHorarioTrabajo._descripcion).First();
    string _horario = _objEn.Encrypt(_id_horario.ToString());
    LogicDetalleHorario _objCatalogoDetalleHorario = new LogicDetalleHorario();
    foreach (var item in _objHorarioTrabajo._listaCrear)
    {
        _objCatalogoDetalleHorario.GuardarDetalleHorario(new entidadDetalleHorario()
        {
            _objHorario= new entidadHorarioTrabajo(){ _id_horario=_horario},
            _objDiaLaborable = new LogicDiaLaborable().ObtenerDiaLaborable
            (int.Parse(_objEn.Decrypt(item._objDiaLaborable._id_dia))),
            _ingreso = item._ingreso,
            _salida = item._salida
        });
    }
}
}

```

**Figura 41.** Método GuardarHorario de LogicHorario.  
**Elaboración:** Autores.

El método *ObtenerHorarios* de la Figura 42 no recibe ningún parámetro y a su vez devuelve una lista de todos los horarios existentes en la BD. El “foreach” recorre los horarios devueltos por el *Sp\_ObtenerHorarios* para añadirlos a la *\_listaHorarioTrabajo* que es devuelta al final del método. Cada horario que compone dicha lista contiene: *\_id\_horario*, mismo que es previamente encriptado, *\_descripción* y una *\_listaDetalle* que se obtiene al consultar, mediante el método *ObtenerDetalleHorario* todos los detalles que correspondan a ese *\_id\_horario*. Dicha lista es ordenada de acuerdo al *\_id\_dia* de los días laborables.

```

/// <summary>
/// Devuelve una lista de todos los horarios administrativos.
/// </summary>
/// <returns>lista de horarios</returns>
1 referencia
public List<entidadHorarioTrabajo> ObtenerHorarios()
{
    List<entidadHorarioTrabajo> _listaHorarioTrabajo = new List<entidadHorarioTrabajo>();
    foreach (var item in _objConexion.Sp_ObtenerHorarios())
    {
        _listaHorarioTrabajo.Add(new entidadHorarioTrabajo()
        {
            _id_horario = _objEn.Encrypt(item.idHorario.ToString()),
            _descripcion = item.descripcion,
            _listaDetalle = new LogicDetalleHorario().ObtenerDetalleHorario(item.idHorario).
                OrderBy(c => c._objDiaLaborable._id_dia).ToList()
        });
    }
    return _listaHorarioTrabajo;
}

```

**Figura 42.** Método *ObtenerHorario* de *LogicHorario*.  
Elaboración: Autores.

En la Figura 43, el método *ActualizarHorarioAgregarDetalle* recibe el *\_objHorarioTrabajo* que contiene el *\_id\_horario* para conocer qué horario se va a modificar y cuáles son los parámetros que se desean agregar. Siendo la tabla *HorarioTrabajo* la primera que se ve afectada mediante el *Sp\_HorarioTrabajo* y finalmente se recorre la *\_listaCrear* que contiene los nuevos detalles que serán agregados a dicho horario mediante el método *GuardarDetalleHorario*. El método que se analiza tiene como fin añadir una nueva hora de entrada y salida a un horario existente. En el sistema existe una variación de este método, que se utiliza para actualizar las horas de ingreso y salida de algún horario existente, pero que mantiene la misma lógica.

```

/// <summary>
/// Actualiza un horario, unicamente hora de entrada y salida de los días guardados.
/// </summary>
/// <param name="_objHorarioTrabajo">objeto horario de trabajo</param>
1 referencia
public void ActualizarHorarioHoras(entidadHorarioTrabajo _objHorarioTrabajo)
{
    _objConexion.Sp_ActualizarHorario(int.Parse(new LogicQueryString().
        Decrypt(_objHorarioTrabajo._id_horario)), _objHorarioTrabajo._descripcion);

    foreach (var item in _objHorarioTrabajo._listaEditar)
    {
        new LogicDetalleHorario().ActualizarDetalleHorario(new entidadDetalleHorario()
        {
            _id_detalle_horario = item._id_detalle_horario,
            _ingreso = item._ingreso,
            _salida = item._salida
        });
    }
}

```

**Figura 43.** Método ActualizarHorario de LogicHorario.  
Elaboración: Autores.

Finalmente, el método *EliminarHorario* de la Figura 44 recibe el *\_id\_horario*, llama al *\_Sp\_EliminarHorarioPorId* para eliminar ese registro de la tabla *HorarioTrabajo* y menciona el método *EliminarDetalleHorarioPorIdHorario* que se encarga de buscar y eliminar todos los detalles relacionados con ese horario. Además, existe una variación de este método, que en lugar de eliminar el horario por completo, solo elimina detalles específicos.

```

/// <summary>
/// Elimina un horario.
/// </summary>
/// <param name="_idHorarioTrabajo">id horario de trabajo</param>
1 referencia
public void EliminarHorario(int _idHorarioTrabajo)
{
    _objConexion.Sp_EliminarHorarioPorId(_idHorarioTrabajo);
    new LogicDetalleHorario().EliminarDetalleHorarioPorIdHorario(_idHorarioTrabajo);
}

```

**Figura 44.** Método EliminarHorario de LogicHorario.  
Elaboración: Autores.

De aquí en adelante, ordenados alfabéticamente según el catálogo, se explican en una tabla todos los métodos, tanto los que pertenecen al CRUD, como los que no. Es preciso mencionar que en cada tabla la columna con el rótulo “A” corresponde al tipo de acceso del método, donde “PB” significa público y “PV” privado.

### 1.13.2 ACCESO

Permite consultar y registrar el acceso que realiza cada usuario al sistema y a su vez utiliza métodos del *LogicUsuario*. Esta clase permite mantener pistas de auditoria, en caso que se requiera realizar controles posteriores. En el Cuadro 97 se muestran los métodos que contiene esta clase.

**Cuadro 97.** Métodos de LogicAcceso.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarAcceso</b>	PB	Guarda un nuevo acceso al sistema (ingreso o salida).	Objeto acceso	"true" o "false".	Si
<b>ObtenerAccesoPorUsuario</b>	PB	Obtiene la lista de accesos de un usuario.	Id del usuario	Lista de accesos.	Si
<b>VerificarDiaYHoraLaborable</b>	PV	Verifica que el día y hora actuales estén en el rango del horario de trabajo del usuario.	Objeto acceso	"true" o "false".	No
<b>ObtenerDiaActual</b>	PV	Traduce el día en español.	Fecha y hora actual.	Día actual en español.	No

Elaboración: Autores.

### 1.13.3 ADJUNTO

Permite gestionar la información no estructurada (documentos) que se tramitan durante la ejecución de los procesos. Almacena los archivos nuevos que suben los usuarios Operadores y a la vez permite guardar nuevas versiones (documentos editados). En el Cuadro 98 se describen los métodos que contiene esta clase.

**Cuadro 98.** Métodos de LogicAdjunto.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerAdjuntoPorIdFormularioElemento</b>	PB	Consulta nombre, ruta, descripción y el id del adjunto, según el id formulario elemento.	Id del formulario	Objeto adjunto	Si
<b>ObtenerAdjuntosInstancia</b>	PB	Obtiene los adjuntos creados en la instancia de un proceso.	Id de la instancia	Lista de adjuntos	Si
<b>GuardarAdjunto</b>	PB	Guarda un archivo adjunto.	Objeto adjunto	-	Si
<b>EliminarAdjunto</b>	PB	Elimina todos los archivos adjuntos pertenecientes a una instancia.	Id de la instancia	-	Si

Elaboración: Autores.

### 1.13.4 ATRIBUTO

El Cuadro 99 describe los métodos que permiten gestionar los atributos de cada objeto flujo, según el estándar BPMN 2.0. Es decir que el usuario puede consultar, ingresar, eliminar y modificar los atributos específicos de cada objeto, mismos que se encuentran detallados en el Anexo 7. Hace referencia a los métodos del *LogicTipoDato* para gestionar el tipo al que pertenece cada atributo.

**Cuadro 99.** Métodos de LogicAtributo

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerAtributosPorElemento</b>	PB	Obtiene los atributos de un elemento específico.	Id del elemento	Lista de atributos	Si
<b>ObtenerAtributos</b>	PB	Obtiene todos los atributos registrados en la BD.	-	Lista de atributos.	Si

Elaboración: Autores.

### 1.13.5 CABECERA TAREA

Gestiona los datos necesarios para crear un nuevo proceso, en cuanto a las fechas de inicio, fecha límite, fecha de finalización, objetivo del proceso, etc. A su vez, utiliza el *LogicTarea* para gestionar todo lo referente a las tareas que componen la cabecera, el *LogicUsuario* para gestionar las actividades del supervisor y del administrador del proceso y el *LogicMensaje* para actualizar y enviar los mensajes para todos los participantes, en caso de modificaciones importantes durante el diseño del proceso. En el Cuadro 100 se presentan los métodos que componen este catálogo.

**Cuadro 100.** Métodos de LogicCabeceraTarea.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ProcesoListo</b>	PB	Establece el proceso en estado "listo" cuando se ha completado con éxito la asignación de recursos.	Id del proceso	Objeto mensaje tarea	Si
<b>GuardarCabeceraTarea</b>	PB	Permite guardar la cabecera y la lista de tareas para la definición de un proceso.	Objeto Cabecera Tarea	Id de la cabecera guardada.	Si
<b>ObtenerCabecerasTareas</b>	PB	Permite consultar una lista de todas las cabeceras de los procesos que no han sido eliminados.	-	Lista de todas las cabeceras que no han sido eliminadas.	Si

<b>ObtenerCabecerasTareasId</b>	PB	Permite consultar los registros (incluyendo usuarios administrador y supervisor), de una cabecera en concreto.	Id de la cabecera	Objeto cabecera tarea.	Si
<b>FiltrarCabecerasTareasId</b>	PB	Permite consultar los registros (excluyendo usuarios administrador y supervisor), de una cabecera en concreto.	Id de la cabecera	Objeto cabecera tarea.	Si
<b>EliminarCabecera</b>	PB	Elimina la cabecera de un proceso y toda la lista de tareas que le pertenecen (eliminado lógico) y a su vez, crea una lista de mensajes que han sido devueltos por el Sp_EliminarCabecera.	id de la cabecera	Lista de mensajes a los diferentes afectados de la tarea eliminada.	Si
<b>ObtenerCabecerasPorIdentityYRol</b>	PB	Permite consultar una lista de los procesos en los que se encuentra el usuario con el rol correspondiente	identity y rol del usuario	Lista de procesos en los que participa el usuario.	
<b>ObtenerCabecerasTareasIdidentityYRol</b>	PB	Permite consultar las cabeceras, según el id cabecera, identity y rol del usuario.	Id cabecera , identity y rol del usuario.	Objeto cabecera con su respectiva lista de tareas, usuario Administrador y usuario Supervisor.	Si
<b>ActualizarFecha</b>	PB	Actualiza la fecha máxima para completar las tareas de la creación de un proceso.	Objeto cabecera	Mensajes para los usuarios que intervienen.	Si
<b>ObtenerCabeceraPorProceso</b>	PB	Consulta la cabecera correspondiente al proceso.	Id del proceso.	Objeto cabecera tarea	Si
<b>RediseñarProceso</b>	PB	Permite establecer el proceso en modo de Rediseño	Id cabecera	-	Si

Elaboración: Autores.

### 1.13.6 CAMPO TAREA

Permite gestionar las etiquetas de las tareas que deben realizar el Modelador, Diseñador, DBA, etc. Por ejemplo, para una tarea del Diseñador existe un campo tarea llamado formulario, que va a ser enviado en el detalle de tarea con el código HTML del formulario. Hace referencia a un método del *LogicTipoDato*, para conocer el tipo de dato del campo. En el Cuadro 101 se presentan los métodos del catálogo.

Cuadro 101. Métodos de LogicCampoTarea.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ObtenerCamposId	PB	Consulta los registros (incluyendo el tipo de dato de cada campo) de un campo en concreto.	Id del campo tarea	Lista de campos.	Si
FiltrarCamposId	PB	Consulta el registro (excluyendo el tipo de dato de cada campo) de un campo en concreto.	id del campo tarea	objeto campo tarea	Si

Elaboración: Autores.

### 1.13.7 CARACTERÍSTICA

Permite gestionar información adicional referente a los usuarios del sistema, por ejemplo el fax de un usuario. Hace referencia a un método del *LogicTipoDato*, para conocer el tipo de dato de las características del usuario. En el Cuadro 102 se presentan los métodos del catálogo.

Cuadro 102. Métodos del LogicCaracterística.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ObtenerCaracteristica	PB	Consulta todas las características disponibles en la BD.	-	Lista de características de los usuarios.	Si
ObtenerCaracteristicaId	PB	Consulta los registros (incluyendo el objeto tipo de dato) de una característica en concreto.	id de la característica.	objeto característica.	Si
FiltrarCaracteristicaId	PB	Consulta los registros (excluyendo el objeto tipo de dato) de una característica en concreto	id característica	objeto característica	Si
GuardarCaracteristica	PB	Envía una nueva característica a la BD.	objeto característica	-	Si
ModificarCaracteristica	PB	Modifica una característica existente.	objeto característica	-	Si
EliminarCarateristica	PB	Elimina los registros de una característica.	id de la característica	-	Si

Elaboración: Autores.

### 1.13.8 CARGO

Gestiona los diferentes cargos administrativos que existen en la ESPAM MFL. A su vez, presenta flexibilidad para agregar nuevos cargos en el transcurso del tiempo. En el Cuadro 103 se presentan los métodos del catálogo.

**Cuadro 103.** Métodos del LogicCargo.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
GuardarCargo	PB	Almacena un nuevo cargo administrativo.	cargo administrativo	-	Si
ObtenerCargos	PB	Consulta todos los cargos registrados en la BD.	-	Lista de cargos	Si
ObtenerCargos	PB	Consulta todos los cargos que tengan al menos un usuario.	-	Lista de cargos	Si
FiltrarCargosId	PB	Consulta el registro de un cargo específico.	id del cargo a consultar	objeto cargo	Si
ActualizarCargo	PB	Actualiza el registro de un cargo específico.	objeto cargo	-	Si
EliminarCargo	PB	Elimina los registros de un cargo específico.	id cargo	-	Si

Elaboración: Autores.

### 1.13.9 CONTROL CABECERA

Gestiona el seguimiento a la creación de un nuevo proceso, permitiendo registrar un historial de las acciones que se realicen sobre la cabecera y los detalles de la lista de tareas que correspondan a esa cabecera. En el Cuadro 104 se presentan los métodos del catálogo.

**Cuadro 104.** Métodos de LogicControlCabecera.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ObtenerControlCabecera	PB	Consulta el registro de un control cabecera específico.	id cabecera	lista de control cabecera	Si

Elaboración: Autores.

### 1.13.10 CONTROL INTEGRACIÓN

Gestiona las incidencias ocurridas durante la ejecución de una Integración Interna o Externa. Permite enviar el código del error que se pueda presentar durante la ejecución una integración y en caso de que se efectúe con éxito, se

enviará el estado correspondiente. En el Cuadro 105 se presentan los métodos del catálogo.

**Cuadro 105.** Métodos de LogicControlIntegración.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarControl</b>	PB	Guarda un nuevo registro de control para una integración.	Objeto control	-	Si

Elaboración: Autores.

### 1.13.11 COORDENADA

Gestiona las coordenadas o ubicaciones de los elementos del diagrama XML, que se registran según la versión del proceso. Este catálogo hace referencia a los métodos del *LogicRegla* para conocer la regla que domina la decisión de esa coordenada (en el caso de las compuertas) y del *LogicElemento* para determinar el elemento al que pertenece esa coordenada. Algunos elementos pueden contener más de una coordenada (las compuertas, que pueden tener un origen y dos o más destinos o un destino y dos o más orígenes). En el Cuadro 106 se presentan los métodos del catálogo.

**Cuadro 106.** Métodos de LogicCoordenada.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerCoordenadaPorIdElemento</b>	PB	Consulta el registro de coordenadas para un elemento	Código del elemento	Lista de coordenadas	Si
<b>ObtenerCoordenadaPorOpcionId</b>	PB	Consulta el registro de una coordenada según su id	Objeto coordenada	Objeto coordenada	Si
<b>ObtenerCoordenadaPorOrigen</b>	PB	Consulta las coordenadas por código de origen (valido para saber las rutas de salida de una compuerta).	Código elemento del origen de la coordenada	Lista de coordenadas.	Si
<b>ObtenerCoordenadaPorIdVersionElemento</b>	PB	Obtiene las coordenadas de un id versión elemento	Id versión elemento	Lista de coordenadas.	Si
<b>GuardarCoordenada</b>	PB	Guarda una nueva coordenada en la BD.	objeto coordenada	Id de la coordenada.	Si
<b>EliminarCoordenada</b>	PB	Elimina los registros de coordenada correspondientes a un id versión elemento.	Id versión elemento	-	Si

Elaboración: Autores.

### 1.13.12 DETALLE HORARIO

Gestiona detalles específicos del horario de trabajo: hora de ingreso y salida, dependiendo de la jornada especificado en la *entidadHorario*. Además, hace referencia a un método del *LogicDiaLaborable*, para establecer el día laborable de cada detalle de horario. En el Cuadro 107 se presentan los métodos del catálogo.

**Cuadro 107.** Métodos del LogicDetalleHorario.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerDetalleHorario</b>	PB	Consulta una lista de detalles, pertenecientes a un horario específico.	id horario	lista de detalle horario	Si
<b>GuardarDetalleHorario</b>	PB	Guarda en la BD un nuevo detalle de horario.	Objeto detalle horario.	-	Si
<b>ActualizarDetalleHorario</b>	PB	Actualiza el registro de un detalle de horario específico.	Objeto detalle horario.	-	Si
<b>EliminarDetalleHorario</b>	PB	Eliminar el registro de un detalle horario específico.	id detalle horario	-	Si
<b>EliminarDetalleHorarioPorIdHorario</b>	PB	Elimina todos los detalles de un horario específico.	id horario	-	Si

Elaboración: Autores.

### 1.13.13 DETALLE REGISTRO

Se encarga de gestionar los registros que se ingresan por los Operadores durante la ejecución de los trámites de la instancia de un proceso y que han sido almacenados en un formulario. En el Cuadro 108 se presentan los métodos del catálogo.

**Cuadro 108.** Métodos de LogicControllIntegración.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarDetalleRegistro</b>	PB	Guarda una lista de detalles de un registro.	Lista detalle registro.	-	Si
<b>ObtenerDetalleRegistro</b>	PB	Consulta la lista de detalles de un registro en concreto.	Id del formulario elemento, id del registro.	Lista detalle registro	Si
<b>ObtenerRegistrosDeFormElemento</b>	PB	Consulta todos los registros almacenados en un	Objeto referencia de formulario	Lista de datos o registros.	Si

		formulario elemento durante un rango de fecha específico.	elemento, id del formulario		
<b>EliminarDetalleRegistro</b>	PB	Elimina los datos de la tabla detalle registro, pertenecientes a un registro en concreto.	Id registro	-	Si

Elaboración: Autores.

### 1.13.14 DETALLE TAREA

Permite que el usuario (Modelador, Diseñador, DBA.) gestione las tareas solicitada por el Administrador, durante la creación de un proceso administrativo. Por ejemplo, en el caso del Modelador, el detalle de la tarea es el diagrama XML correspondiente al campo Diagrama. En este catálogo se instancian los métodos del *LogicMensaje* para retornar los mensajes que serán enviados a los usuarios que intervengan (directa o indirectamente) en la realización de la tarea. En el Cuadro 109 se muestran todos los métodos pertenecientes a esta clase.

**Cuadro 109.** Métodos del LogicDetalleTarea.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>RealizarTarea</b>	PB	Guarda la tarea realizada por el usuario; en caso de que ya exista, actualiza el registro, caso contrario, guarda un campo nuevo y emite una notificación al administrador, indicándole que la tarea ha sido finalizada.	Detalle de la tarea.	Mensaje que se envía	Si
<b>RealizarTareaAMedias</b>	PB	Guarda la tarea realizada por el usuario; en caso de que ya exista, actualiza el registro, caso contrario, guarda un campo nuevo sin emitir notificación al administrador.	Objeto detalle	-	Si
<b>ObtenerDetallePorIdTarea</b>	PB	Permite consultar el detalle de una tarea, y en caso de que sea un archivo HTML, consulta y carga ese documento HTML. El objeto campo tarea.	Id tarea	Lista de detalle tarea.	Si
<b>FiltrarDetallePorIdTarea</b>	PB	Permite consultar el detalle de una tarea, y en caso de que sea un archivo HTML, consulta y carga ese documento HTML. Excluye el objeto campo tarea.	Id tarea	Lista de detalle tarea.	Si

Elaboración: Autores.

### 1.13.15 DÍA LABORABLE

Gestiona los 7 días de la semana, previamente registrados. En el Cuadro 110 se muestran los métodos del catálogo.

Cuadro 110. Métodos del LogicDiaLaborable.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerDiaLaborable</b>	PB	Consulta el registro de un día concreto.	id día	Objeto día laborable	Si

Elaboración: Autores.

### 1.13.16 EJECUCIÓN SUB PROCESO

Gestiona la ejecución de los sub procesos mediante el elemento “actividad de llamada” del diagrama XML. Durante la ejecución de un proceso, permite llamar a un subproceso, ubicando en estado “pausado” al proceso principal hasta que se termine de ejecutar el subproceso. En el Cuadro 111 se muestran los métodos del catálogo.

Cuadro 111. Métodos del LogicEjecuciónSubProceso.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarEjecucionSubproceso</b>	PB	Guarda el registro de la ejecución del subproceso.	Objeto ejecución sub proceso	-	Si
<b>PausarInstancia</b>	PB	Establece la instancia en estado E (espera), cuando se ejecuta una actividad de llamada (Subproceso). Internamente añade un registro al historial de la instancia, indicando la fecha en que se pausó esa instancia, el trámite y objeto donde fue pausada.	Id de la instancia, id del trámite, código del elemento de actividad de llamada.	-	Si
<b>ContinuarInstancia</b>	PB	Establece la instancia en estado J (ejecución), cuando se finaliza una actividad de llamada (Subproceso). Internamente añade un registro al historial de la instancia, indicando la fecha en que se reanudó esa instancia,	Id de la instancia, id del trámite, código del elemento de actividad de llamada.	-	Si

		el trámite y objeto donde fue pausada.			
<b>ObtenerEjecucionSubproceso</b>	PB	Consulta el proceso donde está siendo instanciado un subproceso en concreto.	Id instancia	Objeto ejecución subproceso	Si
<b>VerificarInstanciaSubproceso</b>	PB	Verifica si la instancia pertenece a una ejecución mediante actividad de llamada (Suproceso).	Id instancia	Estado: true (si está siendo instanciado), false (si no está siendo instanciado).	Si
<b>VerificarSubprocesosDelInstancia</b>	PB	Verifica si un proceso contiene subprocesos	Id instancia	Estado: true (si el proceso contiene subprocesos), false (si el proceso no contiene subprocesos)	Si

Elaboración: Autores.

### 1.13.17 ELEMENTO

Gestiona todo lo referente a los elementos de cada versión de un proceso administrativo, es decir, cada parte que compone el flujo (dibujado mediante un diagrama XML y de acuerdo a las directrices del estándar BPMN 2.0) que se debe seguir para ejecutar o instanciar el proceso. En el Cuadro 112 se presentan los métodos del catálogo.

**Cuadro 112.** Métodos del LogicElemento

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerElementoPorId</b>	PB	Consulta el registro de un elemento en concreto	id elemento	objeto elemento	Si
<b>ObtenerElementoPorVersion</b>	PB	Consulta el registro de un elemento según la versión del proceso.	Id versión elemento	Objeto elemento	Si
<b>ObtenerElementoPorCodigo</b>	PB	Consulta el registro de un elemento según su código único generado automáticamente desde el diagramador.	Código único del elemento	Objeto elemento	Si
<b>GuardarElemento</b>	PB	Guarda un elemento en la BD.	objeto elemento	-	Si
<b>ObtenerElementoPorProceso</b>	PB	Consulta todos los elementos pertenecientes a un proceso en concreto.	Id del proceso	Lista de elementos	Si
<b>DefinirTipoCompuerta</b>	PB	Consulta el tipo de compuerta, según el id	Id del elemento	Tipo de compuerta, C (convergente), D	Si

del elemento

(divergente)

Elaboración: Autores.

### 1.13.18 ELEMENTO ATRIBUTO

Gestiona la relación entre las entidades *Atributo* y *VersiónElemento*, permitiendo que se guarde el dato de los atributos de cada elemento del diagrama del proceso. Los atributos fueron definidos según lo indicado por el estándar BPMN 2.0. En el Cuadro 113 se presentan los métodos del catálogo.

Cuadro 113. Métodos del *LogicElementoAtributo*.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarElementoAtributo</b>	PB	Guarda el contenido de los atributos de un elemento.	objeto elemento atributo	Estado: "correcto" o "Error al guardar el dato".	Si
<b>ObtenerAtributosPorIdElemento</b>	PB	Obtiene los atributos según el id versión del elemento	Id versión elemento	Lista de atributos	Si

Elaboración: Autores.

### 1.13.19 ERROR

Gestiona los errores que pueden presentarse durante la ejecución de un proceso administrativo. El núcleo permite que el Supervisor resuelva los inconvenientes enviando diferentes opciones de solución, según el problema que presente la instancia en general y/o alguno de sus trámites.

Utiliza métodos del *LogicTramite* para especificar el trámite en el que ocurrió el incidente; del *LogicInstancia* para establecer como estado "terminado" (T) todos los trámites pendientes de esa instancia, en caso de que sea un "error fatal" y a su vez, permite que la instancia sea "reiniciada" si el Supervisor lo decide; y *LogicMensajeTramite* para gestionar la respectiva notificación al Supervisor. Luego de que se ha ejecutado el error, utiliza el *LogicHistorialInstancia* para guardar el respectivo registro de lo sucedido con la instancia. En el Cuadro 114 se presentan los métodos del catálogo.

Cuadro 114. Métodos del LogicError.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
EjecutarError	PB	Analiza el error encontrado y ubica como terminado los trámites pendientes de la instancia Internamente notifica al usuario Supervisor del problema encontrado y guarda en el historial de esa instancia el evento presentado.	Tipo de elemento que tiene el error, id trámite donde ocurrió el error y elemento donde ocurre el error	-	No
Observaciones	PV	De acuerdo al tipo de elemento, emite el error correspondiente.	Tipo de elemento	String de observación	No

Elaboración: Autores.

### 1.13.20 FORMULARIO

Gestiona los formularios que intervienen durante la ejecución del proceso, los cuales sirven para solicitar y guardar información brindada por parte de los usuarios. Utiliza instancias del *LogicFormularioElemento* para consultar y guardar una lista de elementos HTML que pertenecen a cada formulario. En el Cuadro 115 se presentan los métodos del catálogo.

Cuadro 115. Métodos del LogicFormulario.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ObtenerFormulario	PB	Consulta todos los formularios registrados en la BD.	-	lista de formularios	Si
FiltrarFormulario	PB	Consulta el registro de un formulario específico.	id formulario	objeto formulario	Si
ObtenerFormularioPorCabecera	PB	Obtiene los formularios creados para un proceso específico	id cabecera	lista de formularios	Si
GuardarFormulario	PB	Guarda un formulario una vez que la tarea ha sido aprobada	id tarea	-	Si

Elaboración: Autores.

### 1.13.21 FORMULARIOELEMENTO

Gestiona los elementos HTML que intervienen en cada formulario; permitiendo validar que el formulario contenga elementos, que los mismos estén correctamente ubicados y que contengan los atributos mínimos necesarios. En el Cuadro 116 se presentan los métodos del catálogo.

Cuadro 116. Métodos del LogicFormularioElemento

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerElementosDeFormulario</b>	PB	Consulta una lista de los elementos HTML que tiene un formulario específico.	id formulario	lista de elementos HTML	Si
<b>ObtenerDetallesDeElemento</b>	PB	Consulta el registro de un elemento formulario en concreto.	id formulario elemento	objeto elemento de formulario	Si
<b>ObtenerElementosEspecíficosDeFormulario</b>	PB	Obtiene los elementos de un formulario, con condición específica.	id del formulario	lista de elementos HTML del formulario	Si
<b>CargarHTML</b>	PB	Permite cargar en memoria el HTML y enviarlo para la validación	html en formato string	Estado del html, correcto o incorrecto.	No
<b>HTMLVacio</b>	PV	Permite validar si el html contiene elementos.	documento html	Correcto, si no está vacío. Incorrecto, si está vacío.	No
<b>ValidarHTML</b>	PV	Verfica que todos los elementos del documento HTML estén correctos.	documento HTML	-	No
<b>ValidarElemento</b>	PV	Recibe una lista de elementos HTML y verifica que estén correctos.	lista de elementos HTML	-	No
<b>AgregarElemento</b>	PB	Recibe una lista y los diferentes atributos de un elemento para agregar nuevos elementos a esa lista.	lista del tipo formulario elemento, atributo id, atributo name, etiqueta del elemento, atributo type, atributo _id_integracion, atributo _id_integracion_campo, atributo _mascara, atributo _clase, atributo _tipo_integracion, atributo _value	lista de elementos del formulario	No
<b>GuardarFormularioElemento</b>	PB	Guarda en la BD los elementos HTML del formulario.	id formulario, id tarea	-	Si
<b>ObtenerFormularioPorTarea</b>	PB	Obtiene los elementos de un formulario	id de la tarea	objeto formulario	Si
<b>ObtenerEspecíficosDeFormulario</b>	PB	Obtiene los elementos de un formulario con condición específica.	Id del formulario	Lista de elementos de formulario	SI

Elaboración: Autores.

### 1.13.22 HISTORIAL INSTANCIA

Encargado de mantener el historial de las instancias o ejecución de los procesos. Registra todos los cambios que se puedan presentar en las instancias a través del tiempo, desde que se inicia la ejecución (J), cuando es pausada (E) para ejecutar algún proceso interno (subproceso) y cuando finaliza (F) con éxito. También mantiene el registro si se presenta algún error que requiere que la instancia sea terminada por completo (T). Adicionalmente, permite registrar si una instancia fue reiniciada (R). Utiliza el *LogicInstancia* y el *LogicTrámite* para acceder a la instancia y trámite donde ocurren los eventos mencionados. En el Cuadro 117 se presentan los métodos del catálogo.

**Cuadro 117.** Métodos del LogicHistorialInstancia.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarHistorialInstancia</b>	PB	Guarda el registro cuando se produce un cambio en el estado de la instancia: en ejecución (J), pausado (P), terminado (T), finalizado (F).	Objeto historial instancia	-	Si
<b>ObtenerHistorialErrorPorInstancia</b>	PB	Consulta el historial de errores de cada instancia de un proceso.	Id de instancia	Lista de historial de la instancia	Si
<b>ObtenerHistorialPorInstancia</b>	PB	Consulta todo el historial de cada instancia de un proceso.	id instancia	lista de historial de una instancia	Si
<b>FiltrarElementoHistorialPorInstancia</b>	PB	Consulta todo el historial de cada instancia de un proceso, sin tener en cuenta las relaciones u objetos.	id instancia	lista de historial de una instancia	Si

Elaboración: Autores.

### 1.13.23 HISTORIAL TAREA

Permite llevar un control de todos los cambios que se realicen sobre las tareas durante la creación o diseño de un proceso. En el Cuadro 118 se presentan los métodos del catálogo.

Cuadro 118. Métodos del LogicHistorialTarea

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ActualizarFechaYResponsable	PB	Guarda el cambio del responsable y fecha de entrega de una tarea y notifica a los usuarios correspondientes.	Objeto historial tarea	Lista de mensajes	Si
ObtenerHistorialPorCabecera	PB	Consulta el historial de una tarea, según el id de la cabecera.	Objeto elemento	Lista de historiales de una tarea.	Si

Elaboración: Autores.

### 1.13.24 HORARIO TRABAJO

Gestiona los horarios de los trabajadores administrativos. A su vez, utiliza el *LogicDetalleHorario* para consultar y agregar los detalles: tipo de jornada, hora de inicio, hora de fin y el *LogicDiaLaborable* para establecer los días de esas jornadas. Este catálogo fue explicado con detalle en la sección 1.13.1 **MÉTODOS CRUD**, sin embargo se presentan los métodos en el Cuadro 119.

Cuadro 119. Métodos del LogicHorarioTrabajo

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ObtenerHorarios	PB	Devuelve una lista de todos los horarios administrativos.	-	lista de horarios	Si
ObtenerHorarioPorId	PB	Consulta el registro de un horario específico, con su detalle.	id horario	objeto horario administrativo	Si
GuardarHorario	PB	Guarda un nuevo horario, incluido el detalle.	objeto horario de trabajo	-	Si
ActualizarHorarioHoras	PB	Actualiza un horario, únicamente hora de entrada y salida de los días guardados.	objeto horario de trabajo	-	Si
ActualizarHorarioAgregarDetalles	PB	Actualiza un horario, añade nuevos detalles.	objeto horario trabajo	-	Si
EliminarHorario	PB	Elimina un horario.	id horario de trabajo	-	Si
EliminarDetallesHorario	PB	Elimina detalles específicos de un horario.	objeto horario de trabajo	-	Si

Elaboración: Autores.

### 1.13.25 INSTANCIA

Encargado de gestionar la instancia o ejecución de los procesos. Luego de que se ha diseñado un proceso y se ha realizado la respectiva asignación de

recursos, el usuario Operador responsable del evento de inicio del proceso podrá iniciar la instancia de un proceso en el momento que crea oportuno, o en su defecto, se podrán iniciar automáticamente los que son subprocesos de un proceso mayor. Utiliza el *LogicUsuario* para conocer el usuario que inicia y finaliza la instancia; el *LogicTrámite* para determinar el punto en el que actual de la ejecución del proceso y guardar los trámites necesarios que pertenecen a la instancia; *LogicDetalleRegistro*, *LogicRegistro* y *LogicRegistroExternos* para gestionar los registros de la instancia, incluidos los documentos o adjuntos mediante el *LogicAdjunto*. En el Cuadro 120 se presentan los métodos del catálogo.

**Cuadro 120.** Métodos del LogicInstancia.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarInstancia</b>	PB	Guarda el registro cuando se instancia por primera vez un proceso. Internamente notifica al usuario Supervisor y registra en el historial de la instancia lo sucedido. Además, ingresa los trámites que posteriormente serán realizados por los Operadores.	Objeto instancia	Lista de mensajes	Si
<b>ReiniciarInstancia</b>	PB	Guarda el registro cuando se reinicia la instancia de un proceso, porque existieron errores durante la ejecución. Internamente elimina todo lo relacionado a la instancia y notifica al usuario Supervisor sobre el incidente. Crea los nuevos trámites que serán utilizados en la instancia.	Objeto instancia	Lista de mensajes	Si
<b>ObtenerInstanciasTramite</b>	PB	Obtiene el id de la instancia que corresponde a un trámite específico.	Id trámite	Id instancia	Si
<b>ObtenerUsuarioInstancia</b>		Obtiene el id del usuario que inicia la instancia.	Id instancia	Id usuario que inicia instancia	Si
<b>ObtenerInstancias</b>	PB	Consulta todas las instancias disponibles.	-	lista de instancias	Si
<b>ObtenerInstanciaPorId</b>	PB	Consulta el registro de una instancia específica. Incluyendo la lista de trámites de esa instancia.	Id instancia	Objeto instancia	Si
<b>FiltrarInstanciaPorId</b>	PB	Consulta el registro de una instancia específica.	Id instancia	Objeto instancia	Si

		Excluyendo la lista de trámites de esa instancia.			
<b>ObtenerInstanciaPorIdCabecera</b>	PB	Obtiene una lista de las instancias creadas por el proceso	Id cabecera	Lista de instancias	Si
<b>FinalizarInstancia</b>	PB	Finaliza la Instancia del proceso.	Id instancia	-	Si
<b>EjecutarErrorInstancia</b>	PB	Cuando algo falla durante la ejecución de la instancia, ubica como terminado (T) los trámites restantes.	Id instancia, elemento donde ocurre el error.	-	Si
<b>ContarInstanciasFinalizadas</b>	PB	Consulta el total de instancias finalizadas por cabecera.	Id cabecera	Lista del total de instancias finalizadas que incluye el número de días que tardó en ejecutar cada instancia.	Si
<b>DiasTrabajados</b>	PV	Determina los días laborables existentes en un rango de fecha determinado.	Fecha de inicio, fecha de fin.	Número de días trabajados	No
<b>EliminarSubProcesoPorInstancia</b>	PB	Verifica si un proceso en concreto contiene subprocesos para recorrer y mandar a eliminar la instancia de esos subprocesos y posteriormente envía la instancia del proceso principal para que sea eliminada. En caso de que no tenga subprocesos, únicamente envía la instancia del proceso principal.	Id instancia	-	Si
<b>EliminarInstancia</b>	PB	Elimina por completo la instancia de un proceso, manteniendo únicamente un registro de dicha acción en el historial.	Id instancia.	-	Si

**Elaboración:** Autores.

### 1.13.26 INTEGRACIÓN EXTERNA

Gestiona las integraciones que se hacen a otros sistemas. Mediante la utilización de servicios web brindados por la ESPAM MFL u otras organizaciones, se puede consultar información de Bases de Datos externas al Workflow. Tales datos podrán ser manipulados por los usuarios del sistema, durante la ejecución de los procesos en la forma que crean conveniente. En caso de que los servicios web requieran autenticación, el núcleo permite que se envíen las respectivas credenciales. Además, tiene mecanismos para

facilitar el filtro de la información devuelta por los servicios o recursos web. En el Cuadro 121 se presentan los métodos del catálogo.

**Cuadro 121.** Métodos del LogicIntegraciónExterna

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>CargarHtml</b>	PB	Guarda en memoria el HTML para su posterior verificación.	String del HTML.	objeto integración externa	No
<b>ObtenerXml</b>	PB	Obtiene el XML incrustado en el HTML.	HTML decodificado con HtmlDecode	contador de inicio para obtener la cadena xml	No
<b>LeerXmlEntrada</b>	PV	Lee los parámetros de entrada	Documento XML	-	No
<b>LlenarParametrosEntrada</b>	PV	Llena la lista de los parámetros de entrada.	Lista de elementos	-	No
<b>LeerXmlSalida</b>	PV	Lee los parámetros de salida.	Documento xml	-	No
<b>LlenarParametrosSalida</b>	PV	Llena la lista de los parámetros de salida	Lista de elementos	-	No
<b>GuardarIntegracionExterna</b>	PB	Guarda la integración externa.	Objeto con los datos de la integración	-	Si
<b>ObtenerIE</b>	PB	Obtiene los datos de una integración.	Id de la tarea	Objeto integración externa	Si
<b>ObtenerIntegraciónEld</b>	PB	Obtiene los datos de la integración.	Id de la integración	Objeto integración externa	Si

Elaboración: Autores.

### 1.13.27 INTEGRACIÓN INTERNA

Gestiona los métodos de la integración interna, es decir, las consultas que se hagan a los formularios previamente llenados por los Operadores. En el Cuadro 122 se presentan los métodos del catálogo.

**Cuadro 122.** Métodos del LogicIntegraciónInterna.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarIntegracionInterna</b>	PB	Guarda los elementos de una integración interna	integración a guardar	-	Si
<b>ObtenerIntegracionInternaPorIdTarea</b>	PB	Devuelve la integración interna correspondiente al idTarea.	id tarea	Integración interna	Si
<b>EjecutarIntegracionInterna</b>	PB	Ejecuta la Integración Interna	Objeto integración interna	Objeto integración interna	Si
<b>Recursivo</b>	PV	Método recursivo	Parámetros de	-	No

para validar las  
entradas y salidas.

entrada, salida  
y número de  
registros.

Elaboración: Autores.

### 1.13.28 MENSAJE TAREA

Gestiona la comunicación entre el administrador, supervisor y demás usuarios que intervienen durante la creación de un proceso. Los mensajes enviados entre los usuarios y el sistema pueden ser alertas o simples notificaciones, dependiendo de la naturaleza del mismo. Además, contiene la ruta de la tarea que debe realizar el usuario y la fecha máxima de entrega. Automáticamente se actualiza el estado de los mensajes, según sea el caso: enviado (E), recibido (R), leído (L), eliminado (EL), terminado (T) o reasignado (RA). En el Cuadro 123 se presentan los métodos del catálogo.

Cuadro 123. Métodos del LogicMensajeTarea

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerMensajesPorIdUsuario</b>	PB	Consulta los mensajes los mensajes de un usuario específico.	id usuario, rol usuario	Lista de mensajes	Si
<b>CambiarEstadoMensajeALeido</b>	PB	Establece como "leído" el estado de un mensaje.	id mensaje	-	Si
<b>CambiarEstadoMensajeSupervisor</b>	PB	Establece como "leído" el estado de un mensaje de supervisor.	Id cabecera	-	Si
<b>ObtenerMensajes</b>	PB	Consulta todos los mensajes.	-	Lista de mensajes	Si
<b>AceptarRechazarTarea</b>	PB	Permite que el administrador acepte o rechace una tarea enviada por un usuario. A su vez, si todo el proceso está listo, notifica al administrador para que proceda con la asignación de recursos.	objeto mensaje	Lista de mensajes	Si
<b>ObtenerMensajesNuevos</b>	PB	Consulta los mensajes que no han sido leídos por el usuario.	id identity usuario	Lista de mensajes	Si
<b>ObtenerMensajesAlerta</b>	PB	Consulta los mensajes del tipo alerta de un usuario, que no se hayan eliminado o reasignado.	id usuario identity	Lista de mensajes	Si
<b>ObtenerMensajesPorTarea</b>	PB	Consulta los mensajes correspondientes a una tarea específica.	id tarea	Lista de mensajes	Si
<b>ObtenerMensajesCincoAlerta</b>	PB	Si el usuario no tiene	id identity	Lista de	Si

		disponible alertas creadas en la fecha actual, consulta los 5 últimos mensajes del tipo alerta.	usuario	mensajes	
<b>ModificarMensaje</b>	PB	Actualiza un mensaje del administrador, especificando una nueva fecha límite para la entrega del proceso.	objeto tarea	Lista de mensajes	Si

Elaboración: Autores.

### 1.13.29 MENSAJE TRÁMITE

Gestiona los mensajes que son enviados a los usuarios Operadores, responsables de ejecutar los trámites (tareas) durante la ejecución de los procesos. Además, notifica al Supervisor respectivo la situación actual los trámites pertenecientes a los procesos que tiene a su cargo. En el Cuadro 124 se presentan los métodos del catálogo.

**Cuadro 124.** Métodos del LogicMensajeTrámite

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarMensaje</b>	PB	Guarda un nuevo mensaje para un trámite.	Objeto mensaje trámite	-	Si
<b>EnviarMensaje</b>	PB	Envía los mensajes que corresponden a un EFM (Evento Fin de Mensaje).	Objeto mensaje trámite	-	Si
<b>CrearMensajeSupervisor</b>	PB	Crea un mensaje que será enviado al supervisor para la notificación de la nueva instancia.	Id instancia, id proceso	Objeto mensaje trámite	No
<b>MensajeErrorSupervisor</b>	PB	Crea un mensaje que será enviado al supervisor para la notificación de un error de la instancia.	Id instancia, id proceso, id trámite	Objeto mensaje trámite	No
<b>MensajeReinicioSupervisor</b>	PB	Crea un mensaje que será enviado al supervisor para la notificación del reinicio de una instancia.	Id instancia, id proceso	Objeto mensaje trámite	No
<b>MensajeReinicioTramite</b>	PB	Crea un mensaje que será enviado al operador para la notificación de la nueva fecha límite del trámite.	Id instancia, id proceso, id usuario	Objeto mensaje trámite	No
<b>ObtenerMensajesReinicio</b>	PB	Obtiene una lista de mensajes de un trámite	Id trámite, id instancia	Lista de mensaje	Si

		especifico.		trámite	
<b>ObtenerMensajes</b>	PB	Obtiene una lista de mensajes tramite de un usuario con un rol específico	Identity del usuario y rol del usuario	Lista de mensajes trámites	Si
<b>ObtenerMensajesDeUsuario</b>	PB	Obtiene una lista de mensajes trámite de un usuario.	Identity del usuario	Lista de mensajes trámites	Si
<b>ObtenerMensajesAlertaDeUsuario</b>	PB	Obtiene una lista de mensajes tramite de un usuario que estén en alerta	Identity del usuario	Lista de mensajes trámites	Si
<b>ObtenerMensajesDelInstanciaNuevos</b>	PB	Obtiene los mensajes nuevos creados durante la ejecución de una instancia	Id instancia	Lista de mensajes trámites	Si
<b>CambiarEstadoMensaje</b>	PB	Cambia el estado de un mensaje con un rol específico	Objeto mensaje trámite	-	Si
<b>FinalizarMensaje</b>	PB	Establece el mensaje de fin de un trámite	Objeto mensaje trámite	-	Si
<b>NotificarFinError</b>	PB	Notifica al Supervisor la finalización de una instancia por evento de fin Error.	Id instancia, mensaje	-	Si

Elaboración: Autores.

### 1.13.30 OBJETO FLUJO

Gestiona todo lo referente a los objetos correspondientes a los diagramas que permiten diseñar los flujos de trabajo de una organización, siguiendo el estándar BPMN 2.0. Los elementos utilizados por el sistema han sido previamente definidos de acuerdo al estándar mencionado; los atributos, reglas de conexión y demás parámetros de cada elemento han sido establecidos en el Anexo 4. En el cuadro 125 se presenta el método del catálogo.

**Cuadro 125.** Métodos del LogicObjetoFlujo

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerObjetoFlujoPorId</b>	PB	Consulta el registro de un objeto flujo específico.	id objeto flujo	Objeto flujo	Si

Elaboración: Autores.

### 1.13.31 OBJETO FLUJO ATRIBUTO

Gestiona la relación entre la entidad Objeto Flujo y Atributo, permitiendo registrar el dato correspondiente a cada atributo según el objeto de flujo. En el Cuadro 126 se presenta el método de este catálogo.

**Cuadro 126.** Métodos del LogicObjetoFlujoAtributo

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
ObtenerObjetoAtributo	PB	Obtiene los datos del atributo	id objeto flujo atributo	Objeto flujo atributo	Si

Elaboración: Autores.

### 1.13.32 PARÁMETRO INTEGRACIÓN EXTERNA

Gestiona los parámetros de entrada y salida de la integración externa; permite que el usuario DBA registre los criterios de búsqueda durante la consulta a un servicio web externo al núcleo del Workflow (parámetros de entrada) y facilita el filtro de los resultados obtenidos de dicha búsqueda (parámetros de salida). En el Cuadro 127 se presentan los métodos del catálogo.

**Cuadro 127.** Métodos del LogicParámetroIntegraciónExterna

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
GuardarParametroIE	PB	Guarda los parámetros de entrada de una integración.	parámetros a guardar, id de la integración a la cual pertenecen los parámetros	-	Si
FiltrarParametrosIE	PB	Obtiene los parámetros de salida de una integración	Id de la integración	-	Si

Elaboración: Autores.

### 1.13.33 PARÁMETRO INTEGRACIÓN INTERNA

Gestiona los parámetros de entrada y salida para las integraciones internas; permite que el usuario DBA registre los criterios de búsqueda durante la consulta a los registros ingresados en los formularios del núcleo del Workflow (parámetros de entrada) y facilita el filtro de los resultados obtenidos de dicha búsqueda (parámetros de salida). En el Cuadro 128 se presentan los métodos del catálogo.

**Cuadro 128.** Métodos del LogicParámetroIntegraciónInterna

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarParametrosII</b>	PB	Guarda los parámetros de una integración interna	parámetros a guardar, id de la integración a la cual pertenecen los parámetros.	-	Si
<b>ObtenerParametrosPorIdII</b>	PB	Consulta los parámetros según el id de la integración interna.	id integración interna	lista de parámetros de integración interna	Si

Elaboración: Autores.

### 1.13.34 PROCESO

Gestiona los procesos administrativos de la organización. En el Cuadro 129 se presentan los métodos del catálogo.

**Cuadro 129.** Métodos del LogicProceso.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerProcesos</b>	PB	Consulta todos los procesos existentes en la BD. Incluyendo la lista de versiones de ese proceso.	-	Lista de procesos	Si
<b>ObtenerProcesoPorId</b>	PB	Consulta todos los procesos existentes en la BD. Excluyendo la lista de versiones de ese proceso	-	Lista de procesos	Si
<b>VerificarRediseño</b>	PB	Verifica si el proceso está listo para el rediseño.	Id cabecera	Estado: true, si está listo y false, si no lo está.	Si
<b>ObtenerProcesoPorVersion</b>	PB	Obtiene el id de un proceso, según una versión específica.	Id versión	Id proceso.	Si
<b>ObtenerModeloXCabecera</b>	PB	Obtiene el modelo del proceso	id cabecera, id proceso	Objeto proceso	Si
<b>GuardaProceso</b>	PB	Guardar un proceso	id cabecera	id del proceso creado	Si
<b>ObtenerModeloXCodigoUnico</b>	PB	Obtiene el XML del proceso	Código único de cabecera	XML en string	Si
<b>ObtenerCodigosConInicioMensaje</b>	PB	Consulta todos los códigos de cabecera tarea que estén listos o instanciados, que tenga evento de inicio de mensaje y que la versión esté en true	-	Lista de cabeceras	Si
<b>ObtenerCodigosConFinMensaje</b>	PB	Consulta todos los	-	Lista de	Si

		códigos de cabecera tarea que estén listos o instanciados, que tenga evento de fin de mensaje y que la versión esté en true		cabeceras	
<b>ObtenerProcesosPorIdentity</b>	PB	Obtiene los procesos que inicia el usuario, por identity	Identity	Lista de procesos.	Si
<b>ObtenerProceso9001</b>	PB	Permite consultar la cabecera, según el id de cabecera, para posteriormente generar la ficha ISO 9001.	Id cabecera	Objeto proceso.	Si
<b>EstablecerFechaLimiteProceso</b>	PB	Establece una nueva fecha límite para finalización de un proceso, en modo rediseño.	Objeto cabecera tarea	Estado: true, si se la operación se realizó con éxito y false, si no se pudo completar.	Si
<b>EstablecerFechaLimiteTarea</b>	PB	Establece una nueva fecha límite para una tarea, en modo rediseño.	Objeto tarea	Estado: true, si se la operación se realizó con éxito y false, si no se pudo completar.	Si

Elaboración: Autores.

### 1.13.35 QUERY STRING

Permite encriptar y desencriptar los id de las entidades del sistema. Esta clase ha sido diseñada para mantener la integridad de las claves primarias y foráneas de las entidades, mediante un mecanismo que convierte una matriz de enteros (sin signos) de 8 bits en su representación de cadena equivalente en string, que se codifica con dígitos de base 64. En el Cuadro 130 se presentan los métodos del catálogo.

**Cuadro 130.** Métodos del LogicQueryString

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>Encrypt</b>	PB	Encripta una cadena string.	Cadena no encriptada	Cadena encriptada	Si
<b>Decrypt</b>	PB	Desencripta una cadena string.	Cadena encriptada	Cadena desencriptada	Si

Elaboración: Autores.

### 1.13.36 REFERENCIA FORMULARIO

Gestiona las consultas (referencias) a formularios que han sido llenados por los usuarios Operadores durante la ejecución (instancia) de los procesos. Permite que el Usuario Diseñador pueda mantener el registro de los criterios de búsqueda efectuados a través del tiempo, donde se indica el rango de fechas que se debe analizar para consultar datos específicos y el formulario al que se está haciendo la referencia. En el Cuadro 131 se presentan los métodos del catálogo.

**Cuadro 131.** Métodos del LogicReferenciaFormulario.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarReferenciaFormElemento</b>	PB	Guarda la referencia de ingreso que se le hace a un formulario.	objeto referencia formulario	-	Si
<b>ObtenerReferenciaFormulario</b>	PB	Consulta la referencia de una tarea específica.	id tarea	objeto referencia formulario	Si
<b>EliminarReferencia</b>	PB	Elimina una referencia	id tarea	-	Si
<b>EjecutarReferencia</b>	PB	Permite ejecutar una vista de datos.	Id tramite	objeto referencia formulario	
<b>EjecutarVDF</b>	PB	Permite guardar los datos adjuntos de un registro.	Objeto registro	Estado: true, si se la operación se realizó con éxito y false, si no se pudo completar.	Si

Elaboración: Autores.

### 1.13.37 REFERENCIA FORMULARIO ELEMENTO

Gestiona la relación entre la entidad referencia de formulario y elemento del formulario, además permite especificar si un registro es o no un documento. En el Cuadro 132 se presentan los métodos del catálogo.

**Cuadro 132.** Métodos del LogicReferenciaFormularioElemento.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarReferenciaFormElemento</b>	PB	Guarda un nuevo registro del tipo referencia formulario elemento.	id referencia, id formulario elemento, establece si el archivo es editable	-	Si
<b>FiltrarElementosdeFormulario</b>	PB	Consulta la lista de elementos pertenecientes a una referencia en concreto	id referencia	lista de elementos	Si
<b>EliminarReferenciaFormElemento</b>	PB	Elimina un registro del tipo referencia elemento.	id formulario elemento	-	Si

Elaboración: Autores.

### 1.13.38 REFERENCIA INTEGRACIÓN

Registra la referencia que se hace a las integraciones internas o externas del sistema. En el Cuadro 133 se presentan los métodos del catálogo.

**Cuadro 133.** Métodos del LogicReferenciaIntegración.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarReferenciaIntegración</b>	PB	Guarda la referencia de la integración.	Objeto referencia integración	-	Si
<b>ObtenerReferenciaIntegraciónXTarea</b>	PB	Obtiene los datos de la referencia de la integración por el id de la tarea	Id tarea	Objeto referencia integración	Si

Elaboración: Autores.

### 1.13.39 REGISTRO

Se encarga de gestionar los registros ingresados por los usuarios Operadores durante la ejecución de los trámites de los procesos administrativos. En el Cuadro 134.se presentan los métodos del catálogo.

**Cuadro 134.** Métodos del LogicRegistro.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarRegistroFormulario</b>	PB	Guarda un registro para un trámite.	Objeto registro	Estado: true, si se la operación se realizó con éxito y false, si no se pudo completar.	Si

<b>ObtenerRegistrosPorIdTramite</b>	PB	Obtiene los registros pertenecientes a un trámite	Id trámite	Lista de objetos registro	Si
<b>EliminarRegistro</b>	PB	Elimina todos los registros pertenecientes a un trámite.	Id trámite	Elimina todos los registros pertenecientes a un trámite.	Si

Elaboración: Autores.

### 1.13.40 REGISTRO EXTERNO

Gestiona los registros de entrada de una Integración Externa. En el Cuadro 135 se presentan los métodos del catálogo.

**Cuadro 135.** Métodos del LogicRegistroExterno.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>EliminarRegistrosExternos</b>	PB	Elimina los registros externos pertenecientes a un trámite.	Id trámite	-	Si
<b>EjecutarVDI</b>	PB	Guarda el dato utilizado en una integración.	Lista de registros externos.	Estado: true, si se la operación se realizó con éxito y false, si no se pudo completar.	Si

Elaboración: Autores.

### 1.13.41 REGLA

Permite gestionar las reglas que se utilizan en las compuertas para la toma de decisiones durante la ejecución del flujo de los procesos administrativos. En el Cuadro 136 se presentan los métodos del catálogo.

**Cuadro 136.** Métodos del LogicRegla.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarRegla</b>	PB	Guarda una regla correspondiente a la coordenada del elemento (compuerta)	Objeto regla	-	Si
<b>ObtenerRegla</b>	PB	Obtiene los datos de la regla	Id regla	Objeto regla	Si
<b>ObtenerDatosParaRegla</b>	PB	Obtiene una objeto de regla con los datos necesarios para crearla	Id versión elemento	Objeto regla	Si
<b>ObtenerVersionAnterior</b>	PV	Consulta la versión anterior a la versión actual.	Id versión actual	Id versión anterior	Si
<b>EsTareaAnterior</b>	PV	Recursivo, consulta si el trámite anterior corresponde a una tarea. Se repite las veces que sea necesario para	Objeto trámite	Objeto trámite	No

encontrar su objetivo.					
<b>ComprobarRegla</b>	PB	Ejecuta una regla, perteneciente a una compuerta.	Id versión elemento, id trámite, tipo de compuerta	-	Si
<b>BuscarDefault</b>	PV	Busca si una compuerta contiene una condición por default.	Id versión elemento	Objeto coordinada, nulo si no contiene opción default.	No
<b>EsNum</b>	PV	Verifica si un valor es o no un número.	Valor a comparar	True si es número, false si no lo es.	No
<b>VerificacionRegla</b>	PV	Determina si una regla se cumple, según el criterio de comparación y los valores a comparar.	Lista de reglas, regla, dato a comparar, comparador, id regla.	Lista de reglas con el los id de las reglas que se cumplen.	No

Elaboración: Autores.

### 1.13.42 REPORTE

Gestiona toda la repostería brindada por el sistema. Permite que los usuarios Diseñadores, Supervisores y Administradores puedan diseñar y crear los reportes que necesiten, en cualquier momento. En el Cuadro 137 se presentan los métodos del catálogo.

Cuadro 137. Métodos del LogicReporte.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerReporte</b>	PB	Consulta todos los reportes del sistema.	-	lista de reportes	Si
<b>GuardarReporte</b>	PB	Guarda un nuevo reporte en el sistema.	objeto reporte	-	Si

Elaboración: Autores.

### 1.13.43 RUTA

A partir del id de tarea, ruta base, rol del usuario y el historial, crea una ruta específica para que el usuario responsable (DBA, Diseñador, Modelador, Supervisor...) pueda realizar la tarea que tiene a su cargo. En el Cuadro 138 se presenta el método de este catálogo.

Cuadro 138. Métodos del LogicRuta.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
Ruta	PB	Obtiene la ruta para la realización de las tareas	objeto mensaje tarea	ruta construida	No
RutaTramite	PB	Obtiene la ruta para la realización de los trámites	Objeto de mensaje trámite	Ruta del trámite	No

Elaboración: Autores.

### 1.13.44 TAREA

Gestiona las tareas que deben realizar los usuarios (DBA, Diseñador, Modelador, Supervisor) durante el diseño o creación de un proceso administrativo. En el Cuadro 139 se presentan los métodos del catálogo.

Cuadro 139. Métodos del LogicTarea.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
GuardarTarea	PB	Guarda la creación de nuevas tareas encargadas por el administrador y el respectivo control de la cabecera, a su vez notifica a los usuarios responsables de las mismas, actualiza el progreso de la realización del proceso perteneciente a esas tareas y en caso de que esté al 100% notifica al administrador del mismo.	objeto tarea, id cabecera	-	Si
GuardarNuevaTarea	PB	Guarda la creación de nuevas tareas encargadas por el administrador y el respectivo control de la cabecera, a su vez notifica a los usuarios responsables de las mismas, actualiza el progreso de la realización del proceso perteneciente a esas tareas y en caso de que esté al 100% notifica al administrador del mismo. Además devuelve una lista de los respectivos mensajes.	objeto tarea	lista de mensajes	Si
ObtenerListaTareasPorCabecera	PB	Consulta todas las tareas pertenecientes a un proceso.	id cabecera tarea	lista de tareas	Si
FiltrarTareald	PB	Consulta el registro de una tarea específica, excluyendo el objeto usuario y cabecera y la lista de detalle tarea.	id tarea<	objeto tarea	Si
ObtenerDetalleTarea	PB	Consulta el registro de una tarea específica, incluyendo el objeto usuario y cabecera y la lista de detalle tarea.	id tarea	objeto tarea	Si
EliminarTarea	PB	Elimina el registro de una tarea específica por parte del administrador y notifica al usuario dicho suceso.	id tarea	lista de mensajes	Si
ObtenerTareasPorIdentity	PB	Consulta las tareas de acuerdo al	Identity, rol	lista de	Si

<b>RoIYCabecera</b>		identity, rol e id de la cabecera.	de usuario y id del proceso o cabecera	mensajes	
<b>ObtenerListaIntegraciones</b>	PB	Obtiene la lista de las integraciones aprobadas	-	lista de tareas	Si
<b>ObtenerListaIntegraciones SinEntradaConSalida</b>	PB	Obtiene la lista de las integraciones aprobadas que no tengan parámetros de entrada y que tengan parámetros de salida	-	lista de tareas	Si

Elaboración: Autores.

### 1.13.45 TIPO DE DATO

Gestiona los tipos de datos para los registros de la BD; pueden ser generales como "int", o específicos como "cédula". En el Cuadro 140 se presentan los métodos del catálogo.

**Cuadro 140.** Métodos del LogicTipoDato.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerTipoDato</b>	PB	Consulta todos los tipos de datos disponibles.	-	lista de tipos de dato	Si
<b>FiltrarTipoDatold</b>	PB	Consulta el registro de un tipo de datos específico.	id tipo dato	objeto tipo de dato	Si

Elaboración: Autores.

### 1.13.46 TRÁMITE

Encargado de gestionar los trámites (tareas o pasos) de cada instancia (ejecución) de los procesos. Un trámite corresponde a la ejecución de cada actividad (tarea, evento o compuerta) del flujo de procesos. Puede tener diferentes estados, según las acciones que realice (o deje de realizar) el usuario Operador responsable del mismo; "iniciado" cuando está listo para ser ejecutado, "en ejecución" cuando está siendo realizado por el usuario, "finalizado" cuando termina con éxito o "terminado" cuando termina con algún error. En el Cuadro 141 se presentan los métodos del catálogo.

Cuadro 141. Métodos del LogicTrámite.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerTramitesUsuario</b>	PB	Obtiene una lista de los trámites correspondientes al usuario especificado.	identity del usuario, string de códigos de los procesos	Lista de trámites	Si
<b>ReasignarResponsable</b>	PB	Reasigna el usuario responsable a cada trámite.	Lista de trámites	-	Si
<b>ObtenerTramitesPendientes</b>	PB	Consulta todos los trámites pendientes de un usuario específico.	id del usuario, string de códigos de los procesos	Lista de trámites	No
<b>ObtenerTramitesPendientes PorUsuario</b>	PB	Consulta los trámites pendientes de un proceso que se ejecutan por usuario que instancia donde ese usuario sea responsable de instanciar y consulta los trámites pendientes de un proceso que se ejecutan por usuario responsable	id del usuario, string de códigos de los procesos	Lista de trámites	Si
<b>CalcularTiempoEjecucion</b>	PV	Calcula el tiempo que tomó un trámite en ser ejecutado.	Fecha de inicio, fecha de fin.	Tiempo en días, horas y segundos que tardó en ser ejecutado.	No
<b>ObtenerTramitePorId</b>	PB	Consulta el registro de un trámite específico. Incluyendo el objeto versión elemento.	id trámite	Objeto trámite	Si
<b>FiltrarTramitePorId</b>	PB	Consulta el registro de un trámite específico. Excluyendo el objeto versión elemento.	id trámite	Objeto trámite	Si
<b>ObtenerTramitePorVersionElemento</b>	PB	Consulta el registro de un trámite específico por version	Id versión, id instancia	Objeto trámite	Si
<b>ObtenerTramitePorOpcion InstanciaCoordenada</b>	PB	Consulta el registro de un trámite específico por version	Id instancia, id coordenada	Objeto trámite	Si
<b>ObtenerTramitePorInstancia</b>	PB	Consulta el registro específico de un trámite, según la instancia.	id instancia	lista de trámites	Si
<b>ObtenerTramiteAnterior</b>	PB	Consulta el trámite anterior al trámite actual.	id trámite actual, id de la regla(en caso de que sea compuerta, sino admite nullos) y para las compuertas true en caso de que el siguiente debe ser default	Objeto trámite	Si
<b>ObtenerTramiteSiguiente</b>	PB	Consulta el trámite siguiente al trámite actual.	id trámite actual, id de la regla(en caso de que sea	Objeto trámite	Si

			compuerta, sino admite nulos) y para las compuertas true en caso de que el siguiente debe ser default		
<b>ObtenerDatoPorIdTramite IdFormElemento</b>	PB	Consulta el dato registrado desde un formulario por el usuario.	id trámite donde se realizó el registro, id elemento HTML del formulario	Dato registrado	Si
<b>GuardarTramite</b>	PB	Guarda un trámite	Objeto trámite	-	Si
<b>ReiniciarTramite</b>	PB	Ubica una nueva fecha límite a un trámite, y notifica al operador responsable.	Objeto trámite	Lista de mensajes trámite	Si
<b>FiltrarTramitePorInstancia</b>	PB	Filtra registros específicos de un trámite, según la instancia.	Id instancia	Lista de trámites	Si
<b>EjecutarTramite</b>	PB	Analiza el tipo de elemento y envía para que posteriormente se ejecute un trámite	Trámite a ejecutar, id del trámite siguiente	Lista de mensajes trámite	No
<b>EjecutarCompuerta</b>		Ejecuta el trámite correspondiente a la compuerta (decisión).	Id trámite actual, id trámite siguiente.	-	Si
<b>ObtenerRuta</b>	PV	Construye una ruta para que el Operador ejecute un trámite.	Id trámite	String de la ruta	No
<b>DeterminarTipoElemento</b>	PV	Define las iniciales para los tipos de compuertas y envía el tipo de objeto flujo para las tareas o eventos.	Id versión elemento	String del tipo de elemento	No
<b>EjecutarActividadServicio</b>	PV	Ejecuta una actividad de servicio	Id trámite actual, id trámite siguiente, id instancia, Id versión elemento	-	Si
<b>EjecutarCompuertaParalela</b>	PV	Ejecuta una compuerta paralela	Id trámite siguiente, id instancia	-	Si
<b>EnviarMensajeATSPorCargo</b>	PV	Guarda el mensaje para el usuario Operador, de las actividades por servicio.	Objeto trámite, id versión elemento, id usuario	-	Si
<b>VerificarTareasActivadas YCumplidas</b>	PV	Verifica que todos los trámites estén realizadas en una instancia concreta.	Id versión, id instancia	Estado: true si todo está cumplido, false si hay trámites pendientes	Si
<b>VerificarCumplimientoTareas</b>	PV	Verifica que todos los trámites anteriores estén realizadas para una compuerta	Id versión elemento, id instancia	Estado: true si todo está cumplido, false si hay trámites pendientes	Si
<b>EjecutaTrámite</b>	PV	Ejecuta un trámite.	Id trámite actual, id trámite siguiente, id	-	Si

			instancia, elemento, tipo de elemento, si es compuerta paralela, id versión elemento		
<b>EnviarMensaje</b>	PV	Envía los mensajes correspondientes a los Operadores y Supervisores durante la creación y finalización de los trámites.	Objeto trámite, tipo, id versión elemento	-	Si
<b>EjecutarFinSimple</b>	PV	Ejecuta los eventos del tipo fin simple.	Objeto trámite, elemento, id versión elemento, id trámite actual	-	Si
<b>EjecutarFinMensaje</b>	PV	Ejecuta los eventos del tipo fin de mensaje.	Objeto trámite, elemento, id versión elemento, id trámite actual	-	Si
<b>EjecutarFinMensajeUsuario</b>	PV	Ejecuta los eventos del tipo fin simple.	Id versión elemento, id usuario	-	Si
<b>EjecutarFinMensajeCargo</b>	PV	Ejecuta los eventos de fin, cuando la instancia ha iniciado por cargo.	Id versión elemento	-	Si
<b>EjecutarFinTerminarI</b>	PV	Ejecuta los eventos de fin terminal.	Id instancia	-	Si
<b>EjecutarInicioMensajeProceso</b>	PV	Ejecuta una instancia que inicia con un elemento inicio de mensaje.	Id versión elemento	-	Si

Elaboración: Autores.

### 1.13.47 USUARIO

Gestiona los usuarios que intervienen durante la creación y ejecución del proceso. En el Cuadro 142 se presentan los métodos del catálogo.

**Cuadro 142.** Métodos del LogicUsuario.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarUsuario</b>	PB	Guarda un nuevo usuario	objeto usuario	-	Si
<b>ObtenerUsuariosOperadores</b>	PB	Consulta los usuarios operadores.	-	Lista de operadores	Si
<b>ObtenerUsuarios</b>	PB	Consulta todos los usuarios registrados en la BD.	-	lista de usuarios	Si
<b>ObtenerUsuarioSupervisor</b>	PB	Consulta el supervisor de un proceso.	Id de proceso	Id de usuarios supervisor.	Si
<b>FiltrarUsuariosId</b>	PB	Consulta el registro de un usuario específico.	id usuario	objeto usuario	Si
<b>ObtenerUsuariosIdentity</b>	PB	Consulta el registro de un usuario específico	identity	objeto usuario	Si
<b>ObtenerUsuariosPorRol</b>	PB	Consulta una lista de usuarios, según un rol específico.	rol	lista de usuarios	Si
<b>ObtenerUsuariosPorCargo</b>	PB	Obtiene la lista de usuarios que	Id de cargo	Lista de	Si

		tengan cago especificado		usuarios	
<b>ActualizarUsuario</b>	PB	Actualiza el registro de un usuario.	objeto usuario	-	Si
<b>EliminarUsuario</b>	PB	Elimina el registro de un usuario	id usuario	-	Si

Elaboración: Autores.

### 1.13.48 USUARIO CARACTERÍSTICA

Gestiona la relación entre la entidad usuario y característica. En el Cuadro 143 se presentan los métodos del catálogo.

**Cuadro 143.** Métodos del LogicUsuarioCaracteristica.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>ObtenerCaracteristicaPorUsuario</b>	PB	Consulta las características de un usuario específico.	identity	lista de usuario características	Si
<b>GuardarCaracteristicaUsuario</b>	PB	Guarda la relación entre la entidad usuario y características.	objeto usuario característica	-	Si
<b>ModificarCaracteristicaUsuario</b>	PB	Modifica la relación entre la entidad usuario y características.	objeto usuario característica	-	Si
<b>EliminarCaracteristicaUsuario</b>	PB	Elimina la relación entre la entidad usuario y características.	objeto usuario característica	-	Si

Elaboración: Autores.

### 1.13.49 VERSIÓN

Gestiona el versionamiento de los procesos administrativos. Es decir, permite que se apliquen cambios al diseño del proceso y a su vez que se reutilicen los elementos que se mantienen entre versiones. En el Cuadro 144 se presentan los métodos del catálogo.

**Cuadro 144.** Métodos del LogicVersión.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>GuardarVersionNueva</b>	PB	Guarda una nueva versión de un proceso existente.	Versión del proceso	Id versión	Si
<b>ObtenerVersionesPorProceso</b>	PB	Consulta todas las versiones de un proceso.	id proceso	lista de versiones	Si
<b>ObtenerVersionYTramitePorPorceso</b>	PB	Consulta la versión activa del proceso	Id proceso	Lista de versiones	Si

<b>ObtenerVersionPorId</b>	PB	Consulta una versión específica, según el id versión.	id versión	objeto versión	Si
<b>FiltrarVersionPorId</b>	PB	Consulta una versión específica, según el id versión, sin objetos.	id versión	objeto versión	Si
<b>ObtenerVersionesPorCodigoProceso</b>	PB	Consulta una versión específica, según el código.	Código de proceso	Objeto versión	Si
<b>ObtenerIdVersionPorInstancia</b>	PB	Consulta la versión de la instancia especificada.	Id de instancia	Objeto versión	Si
<b>ObtenerIdVersionPorElemento</b>	PB	Obtiene el id de la versión por elemento	Id versión elemento	Id de la versión	Si

Elaboración: Autores.

### 1.13.50 VERSIÓN ELEMENTO

Gestiona la relación entre la entidad usuario y característica. Además permite analizar el documento XML que contiene el diagrama que sigue el estándar BPMN 2.0 (Anexo 4) durante la creación del proceso. A su vez, valida si todos los elementos del diagrama estén correctamente ubicados y que contengan los parámetros mínimos necesarios y devuelve una lista especificando el estado de cada elemento y su respectiva observación. En el Cuadro 145 se presentan los métodos del catálogo.

Cuadro 145. Métodos del LogicVersiónElemento

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CR UD
<b>ObtenerVersionElementoPorId</b>	PB	Consulta el registro específico de un VersionElemento, según el idVersionElemento.	id versión elemento	objeto versión elemento	Si
<b>ObtenerVersionElementoPorVersion</b>	PB	Consulta el registro específico de un VersionElemento, según el idVersionElemento.	id versión elemento	objeto versión elemento	Si
<b>FiltrarVersionElementoPorId</b>	PB	Permite consultar el registro del versión elemento según el id. Omitiendo ciertos datos del objeto.	id versión elemento	objeto versión elemento	Si
<b>ObtenerVersionDeElemento</b>	PB	Permite consultar el registro del versión elemento según el id.	id versión elemento	objeto versión elemento	Si
<b>GuardarVersionElemento</b>	PB	Permite guardar un registro en la tabla VersionElemento.	objeto versión elemento	-	Si
<b>ModificarVersionElemento</b>	PB	Guarda el usuario y el tiempo asignados al elemento	objeto versión elemento	-	Si

<b>EnviarXML</b>	PB	Analiza el XML antes de guardarlo en la BD.	documento xml	-	No
<b>DefinirIdObjFlujo</b>	PV	Para definir el idObjetoFlujo correspondiente a un evento, analizando el tipo de elemento que es según sus atributos.	elemento XML del tipo evento, tipo: "I" inicio, "F" fin	id objeto flujo	No
<b>DefinirTipoCompuerta</b>	PV	Define si una compuerta es convergente "C" o divergente "D".	elemento XML del tipo compuerta, administrador de espacios del XML	tipo: "C" convergente, "D" divergente.	No
<b>EnviarEvento</b>	PV	Recibe una lista de eventos y el tipo específico para almacenarlo en la BD. Utiliza el método DefinirIdObjetoFlujo.	lista de elementos XML del tipo evento, tipo: "I" para inicio y "F" para fin, todos los sequence flow del diagrama XML	-	No
<b>EnviarTarea</b>	PV	Recibe una lista de tareas, el idObjFlujo y el tipo específico para almacenarlo en la BD.	lista de elementos XML del tipo tarea, id objeto flujo, tipo de tarea, lista de elementos XML del tipo ruta (sequenceFlow), id version	-	No
<b>EnviarCompuerta</b>	PV	Recibe una lista de compuertas, el idObjFlujo para almacenarlo en la BD. Utiliza el método DefinirTipoCompuerta	lista de elementos XML del tipo compuerta, id objeto flujo, lista de elementos XML del tipo ruta (sequenceFlow), id versión		No
<b>BuscarCodOrigen</b>	PV	Permite conocer el código origen de un elemento.	lista de elementos XML del tipo ruta (sequenceFlow), entrada del elemento XML, id elemento	colección de atributos correspondientes al código origen<	No
<b>BuscarCodDestino</b>	PV	Permite conocer el código destino de un elemento.	lista de elementos XML del tipo ruta (sequenceFlow), salida del elemento XML, id elemento	colección de atributos correspondientes al código destino	No
<b>EnviarElemVersion</b>	PB	Permite enviar a la BD el elemento versión.	id coordenada, id elemento, id versión	-	No
<b>EnviarCoordenada</b>	PV	Permite enviar a la BD las coordenadas de un elemento.	código del elemento que se envía, origen del elemento, salida del elemento y en caso de compuertas, las opciones para cada ruta, id de la versión del elemento	id coordenada	No
<b>EnviarElemento</b>	PB	Permite enviar a la BD el elemento.	id elemento, descripción, tipo de elemento, objeto flujo	-	No
<b>ValidarXML</b>	PB	Verifica si todos los elementos del XML están correctos.	XML en string	lista de elementos con su respectivo estado y observación	No
<b>AnalizarXML</b>	PB	Verifica que los elementos del XML estén relacionados correctamente, antes de que se envíe el XML a la BD.	documento xml	lista de elementos con su respectivo estado y observación	No

<b>AnalizarEvento</b>	PV	Analiza que las entradas y salidas estén correctas de acuerdo al tipo de evento, y que cada evento contenga los elementos necesarios.	lista de elementos XML del tipo evento, tipo: "I" para inicio y "F" para fin, todos los sequence flow del diagrama XML, administrador de espacios del XML	-	No
<b>AnalizarTarea</b>	PV	Analiza que las entradas y salidas estén correctas de acuerdo al tipo de tarea, y que cada tarea contenga los elementos necesarios.	lista de elementos XML del tipo tarea, todos los sequence flow del diagrama XML, administrador de espacios del XML	-	No
<b>AnalizarTareaLlamada</b>	PV	Analiza que las entradas y salidas estén correctas de acuerdo al tipo de tarea de llamada, y que cada tarea contenga los elementos necesarios.	lista de elementos XML del tipo tarea, todos los sequence flow del diagrama XML, administrador de espacios del XML	-	No
<b>AnalizarCompuerta</b>	PV	Analiza que las entradas y salidas estén correctas de acuerdo al tipo de compuerta, y que cada compuerta contenga los elementos necesarios.	lista de elementos XML del tipo compuerta, todos los sequence flow del diagrama XML, administrador de espacios del XML	-	No
<b>LlenarLista</b>	PV	Agrega los elementos a la lista que será enviada a la interfaz.	id elemento, estado: correcto o incorrecto, explicación del estado establecido	-	No

Elaboración: Autores.

### 1.13.51 WEB SERVICE

Gestiona la conexión con otros servicios web, para las integraciones externas. En el Cuadro 146 se presentan los métodos del catálogo.

**Cuadro 146.** Métodos del Webservice.

Método	A	Descripción	Parámetro(s) que recibe.	Parámetro que retorna.	CRUD
<b>Invoke</b>	PB	Invokes service	Added parameters will encode? (default: true), autenticación	llama al método invoke, mismo que devuelve un objeto integración externa	No
<b>Invoke</b>	PB	Invokes service	Added parameters will encode? (default: true), autenticación, Lista de parámetros	objeto integración externa	No
<b>LeerXmlSalida</b>	PV	Lee el XML de salida	documento xml, lista de parámetros	-	No
<b>LlenarRegistros</b>	PV	Llena la lista de los registros por parámetros de salida	lista de elementos XML	-	No

Elaboración: Autores.

## 1.14 RECURSOS WEB (CONTROLADORES)

A continuación, se detalla el flujo de los recursos de cada controlador, donde se recibe mediante la API web la petición enviada por parte del cliente, posteriormente se envía al catálogo correspondiente que utiliza la respectiva entidad y el procedimiento almacenado para conectarse con la BD. Cabe destacar que al ser una API RestFul únicamente utiliza los métodos estandarizados.

Sin embargo, es necesario establecer que cada controlador hereda de ApiController, la clase específica dentro de ASP.NET Web API para los controladores de una API de recursos web. En la Figura 45 se muestra un ejemplo de herencia con la clase ProcesoController.

```
public class ProcesoController : ApiController
{
```

**Figura 45.** Ejemplo de herencia de ApiController.  
**Elaboración:** Autores

Además, cada controlador se ubica con el atributo [Authorize], ya que se autentican y filtran las peticiones HTTP que se realicen a la API desde las aplicaciones clientes y solo se permite el acceso a los diferentes recursos si la petición la realiza un usuario autorizado. De esta manera, no sólo se consigue autenticar la petición, sino optimizar el uso de recursos del servidor, ya que la petición no llega al controlador ni se ejecuta su código si la autenticación realizada sobre el análisis de las cabeceras de la petición HTTP no arroja un resultado satisfactorio.

### 1.14.1 ACCESO

Permite llevar un registro o control del acceso al sistema, por parte de todos los usuarios registrados en la BD del sistema. En el Cuadro 147 se presenta un resumen de los métodos HTTP que pertenecen al controlador. A modo de ejemplo, a continuación se explican detalladamente los métodos de este controlador.

**Cuadro 147.** Métodos de AccesoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene la lista de accesos (entradas y salidas) de un usuario.	Id del usuario del cual se obtienen los accesos.	Lista de acceso	api/Acceso/5
Post	Guarda un registro cuando el usuario ingresa o sale del sistema.	Datos del acceso	-	api/Acceso

Fuente: Autores.

- **GET:** Obtiene la lista de accesos (entradas y salidas) de un usuario. En la Figura 46 se especifica que se recibe *id* del usuario al que se le van a consultar los accesos registrados, se instancia un objeto del *LogicQueryString* para descifrar dicho id y almacenarlo en la variable temporal *\_id*, luego se crea una nueva lista de accesos que se llena partir del método *ObtenerAccesoPorUsuario* que se encarga de realizar la respectiva consulta, según el *\_id* enviado.

```
public List<entidadAcceso> Get(string id)
{
    LogicQueryString _objQuery = new LogicQueryString();
    int _id = int.Parse(_objQuery.Decrypt(id));
    List<entidadAcceso> _lista_accesos = new List<entidadAcceso>();
    _lista_accesos = new LogicAcceso().ObtenerAccesosPorUsuario(_id);
    return _lista_accesos;
}
```

**Figura 46.** Código del método GET de AccesoController.

Fuente: Autores.

- **POST:** Permite registrar la información básica del usuario, la dirección IP desde la que está accediendo, la dirección MAC del equipo que se

registra, fecha y hora y el tipo de acceso: ingreso o salida del sistema, a fin de contribuir con información relevante para controles o auditorías posteriores. En la Figura 47 se ilustra que el método recibe el objeto del tipo *entidadAcceso* para enviarlo mediante el método *GuardarAcceso* que se llama a partir de la instancia del *LogicAcceso*.

```

public bool Post(entidadAcceso _objAcceso)
{
    LogicAcceso _objCatalogoAcceso = new LogicAcceso();
    try
    {
        return _objCatalogoAcceso.GuardarAcceso(_objAcceso);
    }
    catch (Exception)
    {
        return false;
    }
}

```

Figura 47. Código del método POST de AccesoController.

Fuente: Autores.

### 1.14.2 ACTUALIZAR HTML

Actualiza el archivo HTML que corresponde a alguna tarea realizada por el Diseñador del sistema. Para lo cual utiliza un solo método presentado en el Cuadro 150.

Cuadro 148. Métodos HTTP de ActualizarHtmlController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Actualiza el archivo HTML.	Objeto detalle tarea	-	API/ActualizarHtml

Fuente: Autores.

### 1.14.3 APROBAR TAREA

Permite que el Administrador “apruebe” o “rechace” una tarea realizada y enviada por otro usuario. El Cuadro 149 presenta un resumen del recurso o método HTTP que contiene el controlador.

**Cuadro 149.** Métodos HTTP de AprobarTareaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Aprueba o rechaza una tarea y emite una notificación al administrador y al usuario responsable.	objeto mensaje	lista de mensajes	api/AprobarTarea

Fuente: Autores.

#### 1.14.4 ASIGNAR RECURSOS

Permite asignar los recursos necesarios: diagramas, formularios, integraciones, etc. a los elementos que componen el diagrama XML del proceso. El Cuadro 150 presenta un resumen de los métodos HTTP que contiene el controlador.

**Cuadro 150.** Métodos HTTP de AsignarRecursosController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene el modelo y los elementos del proceso.	id de la cabecera	objeto proceso	API/AsignarRecurso/5

Fuente: Autores.

#### 1.14.5 ATRIBUTO

Permite gestionar los atributos que forman parte de los elementos del diagrama de flujo de los procesos administrativos. Estos atributos han sido previamente definidos según el estándar BPMN 2.0. (Anexo 4). A continuación, en el Cuadro 151 se presentan los métodos HTTP de este controlador.

**Cuadro 151.** Métodos HTTP de AtributoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de todos los atributos.	-	Lista de atributos	API/Atributo
Get	Obtiene una lista de los atributos de un elemento específico.	Id elemento	Lista de atributos	API/Atributo/5

Fuente: Autores.

#### 1.14.6 CABECERA

Permite gestionar los atributos que pueden contener los elementos del diagrama XML. Estos atributos han sido previamente definidos según el

estándar BPMN 2.0. (Anexo 4). A continuación, en el cuadro 152 se presentan los métodos HTTP de este controlador.

**Cuadro 152.** Métodos HTTP de CabeceraController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de todas las cabeceras de tareas	-	Lista de cabeceras	API/Cabecera
Get	Obtiene una cabecera	Id cabecera	Objeto cabecera	API/Cabecera/5
Post	Guarda la cabecera y las tareas que se necesitan para crear un proceso.	Objeto cabecera	Lista de mensajes tarea	API/Cabecera
Put	Establece que el proceso está listo.	Objeto cabecera	Lista de mensajes tarea	API/Cabecera/5
Delete	Elimina una cabecera	Id cabecera	Lista de mensajes tarea	API/Cabecera/5

Fuente: Autores.

### 1.14.7 CARACTERÍSTICA

Permite gestionar las características de los usuarios. A continuación, en el Cuadro 153 se presentan los métodos HTTP de este controlador.

**Cuadro 153.** Métodos HTTP de CaracterísticaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de las características guardadas en la BD	-	Lista de características	API/Característica
Get	Obtiene el detalle de una característica	Id característica	Objeto característica	API/Característica/5
Post	Guarda una nueva característica.	Objeto característica	-	API/Característica
Put	Modifica una característica existente.	Objeto característica	-	API/Característica/5
Delete	Elimina una característica.	Id característica	-	API/Característica/5

Fuente: Autores.

### 1.14.8 CARACTERÍSTICA USUARIO

Permite gestionar los datos o registros de las características de los usuarios. Estos atributos han sido previamente definidos según el estándar BPMN 2.0.

(Anexo 4). A continuación, en el Cuadro 154 se presentan los métodos HTTP de este controlador.

**Cuadro 154.** Métodos HTTP de CaracterísticaUsuarioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Post</b>	Guarda una nueva característica a un usuario.	Objeto característica usuario	-	API/Característica Usuario
<b>Put</b>	Modifica una característica del usuario.	Objeto característica usuario	-	API/Característica Usuario /5
<b>Delete</b>	Elimina una característica de un usuario.	Id característica usuario	-	API/Característica Usuario /5

Fuente: Autores.

### 1.14.9 CARGO

Permite gestionar cargos administrativos de los usuarios Operadores del sistema. A continuación, en el Cuadro 155 se presentan los métodos HTTP de este controlador.

**Cuadro 155.** Métodos HTTP de CargoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Get</b>	Obtiene una lista de cargos.	-	Lista de cargos	API/Cargo
<b>Post</b>	Guarda un cargo nuevo.	Objeto cargo	-	API/Cargo
<b>Put</b>	Modifica un cargo existente.	Objeto cargo	-	API/Cargo/5
<b>Delete</b>	Elimina un cargo.	Objeto cargo	-	API/Cargo/5

Fuente: Autores.

### 1.14.10 CONTAR INSTANCIA

Consulta el total de instancias de un proceso administrativo concreto, según el criterio de consulta. A continuación, en el Cuadro 156 se presentan los métodos HTTP de este controlador.

**Cuadro 156.** Métodos HTTP de ContarInstanciaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Get</b>	Consulta el total de instancias finalizadas.	Id cabecera	Lista de instancias	API/ContarInstancia

Fuente: Autores.

### 1.14.11 CONTROL CABECERA

Gestiona las novedades o acciones importantes que se realizan durante la creación o diseño de un proceso, por ejemplo, cuando se elimina una tarea, se aplaza la fecha límite para finalizar el proceso, etc. A continuación, en el Cuadro 157 se presentan los métodos HTTP de este controlador.

**Cuadro 157.** Métodos HTTP de ControlCabeceraController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de las entradas de control de la cabecera	Id cabecera	Lista de controles cabecera	API/ControlCabecera

Fuente: Autores.

### 1.14.12 DEFINIR COMPUERTA

Define el tipo de compuerta (convergente o divergente) de un elemento de decisión del diagrama o flujo de trabajo de los procesos y facilita la lista de coordenadas del elemento, en caso de que sea divergente. A continuación, en el Cuadro 158 se presentan los métodos HTTP de este controlador.

**Cuadro 158.** Métodos HTTP de DefinirCompuertaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Define el tipo de Compuerta, si es Divergente, se envía la lista de coordenadas.	Id elemento	Lista de coordenadas	API/DefinirCompuerta

Fuente: Autores.

### 1.14.13 DETALLE FORMULARIO

Gestiona los elementos que contienen los formularios HTML creados por el Diseñador y posteriormente registrados por los Operadores. A continuación, en el Cuadro 159 se presentan los métodos HTTP de este controlador.

**Cuadro 159.** Métodos HTTP de DetalleFormularioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de los elementos de un formulario	Id formulario	Lista de formulario elemento	API/DetalleFormulario/5

Fuente: Autores.

#### 1.14.14 DETALLE TAREA

Controla los registros de las tareas de los usuarios Diseñador, Modelador, DBA. A continuación, en el Cuadro 160 se presentan los métodos HTTP de este controlador.

**Cuadro 160.** Métodos HTTP de DetalleTareaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Guarda el detalle o registro de las tareas realizadas, sin notificar al administrador.	Objeto detalle tarea	En caso de que sea una tarea del Diseñador, devuelve una lista de los elementos del XML con un estado (correcto/incorrecto) de cada elemento y una pequeña observación sobre cada elemento. En caso de que sea otro tipo de tarea, guarda el registro de la tarea y devuelve una lista vacía de elementos.	API/DetalleTarea
Put	Guarda el detalle o registro de las tareas realizadas, con notificación al administrador.	Objeto detalle tarea	Lista de mensaje tarea.	API/DetalleTarea/5

Fuente: Autores.

#### 1.14.15 DOCUMENTACIÓN 9001

Permite crear una ficha (documento PDF) basada en la norma 9001, que contiene los detalles más relevantes del proceso. A continuación, en el Cuadro 161 se presentan los métodos HTTP de este controlador.

**Cuadro 161.** Métodos HTTP de Documentación9001Controller.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Permite crear una documentación con la norma 9001.	Id cabecera	Objeto proceso	API/Documentacion9001/1

Fuente: Autores.

### 1.14.16 DOCUMENTOS INSTANCIA

Permite gestionar todos los registros obtenidos a partir de una integración interna. A continuación, en el Cuadro 162 se presentan los métodos HTTP de este controlador.

**Cuadro 162.** Métodos HTTP de DocumentosInstanciaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los registros de formulario.	Id tarea, fecha desde, fecha hasta.	Objeto integración interna	API/DocumentosInstancia/5

Fuente: Autores.

### 1.14.17 EJECUCIÓN ATM

Permite ejecutar las Actividades de Tareas Manuales (ATM) de los diagramas de flujo de los procesos administrativos. A continuación, en el Cuadro 163 se presentan los métodos HTTP de este controlador.

**Cuadro 163.** Métodos HTTP de EjecuciónATMController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Ejecuta el trámite correspondiente a la tarea manual	Id trámite	Lista de mensajes trámite	API/EjecuciónATM/5

Fuente: Autores.

### 1.14.18 EJECUCIÓN FORMULARIO

Gestiona los ingresos (registros) de un formulario durante la ejecución del proceso. El ingreso de la información solicitada en cada formulario es realizado por los Operadores. A continuación, en el Cuadro 164 se presentan los métodos HTTP de este controlador.

**Cuadro 164.** Métodos HTTP de EjecuciónFormularioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Guarda los datos del formulario	Objeto registro	Lista mensaje trámite	API/EjecuciónFormulario

Fuente: Autores.

### 1.14.19 EJECUCIÓN VDF COMPUERTA

Ejecuta la vista de datos de formulario, en una compuerta. A continuación, en el Cuadro 165 se presentan los métodos HTTP de este controlador.

**Cuadro 165.** Métodos HTTP de EjecuciónVDFCompuertaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Guarda los trámites de Vista de Datos Formulario en una compuerta.	Objeto registro	Lista de mensajes trámite	API/EjecuciónVDFCompuerta

Fuente: Autores.

### 1.14.20 EJECUCIÓN VDF

Ejecuta la vista de datos de formulario. A continuación, en el Cuadro 166 se presentan los métodos HTTP de este controlador.

**Cuadro 166.** Métodos HTTP de EjecuciónVDFController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Guarda los trámites de Vista de Datos Formulario.	Objeto registro	Lista mensajes trámite	API/EjecuciónVDF

Fuente: Autores.

### 1.14.21 EJECUCIÓN VDI COMPUERTA

Ejecuta una VDI (Vista de Datos de Integración) de una compuerta. A continuación, en el Cuadro 167 se presentan los métodos HTTP de este controlador.

**Cuadro 167.** Métodos HTTP de EjecuciónVDICompuertaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Guarda los trámites de Vista de Datos Formulario en una compuerta.	Objeto trámite	Lista mensajes trámite	API/EjecucionVDICompuerta/5

Fuente: Autores.

### 1.14.22 EJECUCIÓN VDI

Ejecuta el trámite de una VDI (Vista de Datos de Integración). A continuación, en el Cuadro 168 se presentan los métodos HTTP de este controlador.

**Cuadro 168.** Métodos HTTP de EjecuciónVDIController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Ejecuta la VDI.	Objeto trámite	Estado: correcto, incorrecto	API/EjecuciónVDI
Put	Guarda los trámites de Vista de Datos Integración.	Objeto trámite	Lista de mensajes trámite	API/EjecuciónVDI /5

Fuente: Autores.

### 1.14.23 ELEMENTO

Permite recibir y validar los elementos que conforman los formularios (documentos HTML) elaborados por el usuario Diseñador. A continuación, en el Cuadro 169 se presentan los métodos HTTP de este controlador.

**Cuadro 169.** Métodos HTTP de ElementoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Permite cargar el documento HTML y enviarlo para la validación.	Objeto detalle tarea	True, si los elementos del documento son correctos. False, si son incorrectos.	API/ElementoController

Fuente: Autores.

### 1.14.24 FILTRO ADJUNTO

Filtra los documentos adjuntos, con todos sus atributos, que pertenecen a una instancia en concreto registrada en la BD del sistema. A continuación, en el Cuadro 170 se presentan los métodos HTTP de este controlador.

**Cuadro 170.** Métodos HTTP de FiltroAdjuntoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Filtra los archivos adjuntos que corresponden a una instancia.	Id instancia	Lista de adjuntos	API/FiltroAdjunto/5

Fuente: Autores.

### 1.14.25 FITLRO AREA CABECERA

Permite manejar los procesos administrativos pertenecientes a un usuario y rol concreto, según el estado (pendiente, terminado, instanciado) de los mismos. A continuación, en el Cuadro 171 se presentan los métodos HTTP de este controlador.

**Cuadro 171.** Métodos HTTP de FiltroAreaCabeceraController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene la lista de cabeceras en las que esta designado un usuario.	Identity del usuario, rol del usuario, estado de la cabecera	Lista de cabeceras (procesos)	API/FiltroAreaCabecera/5

Fuente: Autores.

### 1.14.26 FILTRO AREA SUPERVISOR

Permite gestionar los procesos que han sido asignados al Supervisor. A continuación, en el Cuadro 172 se presentan los métodos HTTP de este controlador.

**Cuadro 172.** Métodos HTTP de FiltroAreaSupervisorController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Consulta los procesos asignados al usuario.	Identity del usuario	Lista de procesos.	API/FiltroAreaSupervisor/5

Fuente: Autores.

### 1.14.27 FILTRO AREA TAREA

Permite gestionar las tareas asignadas a un usuario (Modelador, DBA, Diseñador) durante el diseño de un proceso específico. A continuación, en el Cuadro 173 se presentan los métodos HTTP de este controlador.

**Cuadro 173.** Métodos HTTP de FiltroAreaTareaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene la cabecera con la lista de tareas del usuario.	Identity del usuario, rol del usuario, id de la cabecera del proceso.	Objeto cabecera tarea	API/FiltroAreaTarea/5

Fuente: Autores.

### 1.14.28 FILTRO CABECERA

Permite gestionar las cabeceras de los procesos administrativos, que contiene la información base: objetivo, fecha de inicio, fecha de finalización, estado, etc., y la lista de tareas establecidas durante la fase de diseño del proceso. A continuación, en el Cuadro 174 se presentan los métodos HTTP de este controlador.

**Cuadro 174.** Métodos HTTP de FiltroCabeceraController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de todas las cabeceras que contienen tareas terminadas o pendientes.	Tipo de filtro	Lista de cabeceras	API/FiltroCabecera/L
Put	Establece el proceso es estado rediseño.	Objeto cabecera tarea	-	API/FiltroCabecera/objeto

Fuente: Autores.

### 1.14.29 FILTRO CARGO

Gestiona el filtro de los cargos administrativos de la ESPAM MFL. A continuación, en el Cuadro 175 se presentan los métodos HTTP de este controlador.

**Cuadro 175.** Métodos HTTP de FiltroCargoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Consulta todos los cargos registrados en la BD que tengan usuarios.	-	Lista de cargos	API/FiltroCargo

Fuente: Autores.

### 1.14.30 FILTRO FORMULARIO

Gestiona el filtro de los formularios elaborados por el Diseñador durante la creación del proceso y llenados por los Operadores durante la ejecución del mismo. A continuación, en el Cuadro 176 se presentan los métodos HTTP de este controlador.

**Cuadro 176.** Métodos HTTP de FiltroFormularioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de los formularios.	-	Lista de formularios.	API/ FiltroFormulario
Get	Obtiene una lista de los formularios por cabecera.	Id cabecera	Lista de formularios.	API/FiltroFormulario/5

Fuente: Autores.

### 1.14.31 FILTRO INSTANCIA

Permite gestionar el filtro de las instancias de los procesos administrativos. A continuación, en el Cuadro 177 se presentan los métodos HTTP de este controlador.

**Cuadro 177.** Métodos HTTP de FiltroInstanciaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene las instancias correspondientes al id ingresado.	Id de la cabecera del proceso	Lista de instancias	API/FiltroInstancia/5

Fuente: Autores.

### 1.14.32 FILTRO INTEGRACIÓN

Gestiona el filtro de las integraciones internas del workflow. A continuación, en el Cuadro 178 se presentan los métodos HTTP de este controlador.

**Cuadro 178.** Métodos HTTP de FiltroIntegraciónController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de las tareas de tipo integración.	-	Lista de tareas.	API/FiltroIntegración
Get	Obtiene un objeto de tipo formulario con sus elementos	Id de la tarea	Objeto formulario	API/FiltroIntegración/5

Fuente: Autores.

### 1.14.33 FILTRO INTEGRACIÓN FORMULARIO

Gestiona el filtro de las integraciones de los procesos administrativos, según criterios específicos. A continuación, en el Cuadro 179 se presentan los métodos HTTP de este controlador.

**Cuadro 179.** Métodos HTTP de FiltroIntegraciónFormularioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene la lista de las integraciones aprobadas que no tengan parámetros de entrada y que tengan parámetros de salida.	-	Lista de tareas.	API/FiltroIntegraciónFormulario

Fuente: Autores.

### 1.14.34 FILTRO MENSAJE

Permite gestionar el filtro de mensajes nuevos del sistema de los usuarios encargados de diseñar el proceso. A continuación, en el Cuadro 180 se presentan los métodos HTTP de este controlador.

**Cuadro 180.** Métodos HTTP de FiltroMensajeController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los mensajes de un usuario.	Identity del usuario	Lista de mensaje tarea	API/FiltroMensaje

Fuente: Autores.

### 1.14.35 FILTRO MENSAJE TRÁMITE

Encargado del filtro de los mensajes emitidos por el sistema durante la etapa de ejecución de los procesos administrativos a los usuarios Operadores. A

continuación, en el Cuadro 181 se presentan los métodos HTTP de este controlador.

**Cuadro 181.** Métodos HTTP de FiltroMensajeTramiteController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Get</b>	Obtiene los mensajes de trámite de un usuario	Identity del usuario	Lista de mensaje trámite	API/FiltroMensajeTramite/5
<b>Post</b>	Obtiene las notificaciones (del tipo alerta) de un usuario.	Objeto usuario	Lista de mensaje trámite	API/FiltroMensajeTramite
<b>Put</b>	Cambia el estado de un objeto mensaje	Objeto mensaje trámite	-	API/FiltroMensajeTramite/5

Fuente: Autores.

### 1.14.36 FILTRO PROCESO

Permite la gestión del filtro de los procesos administrativos que tienen un diagrama de flujo que tiene un evento de inicio de mensajes o un evento de fin de mensaje. A continuación, en el Cuadro 182 se presentan los métodos HTTP de este controlador.

**Cuadro 182.** Métodos HTTP de FiltroProcesoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Get</b>	Consulta todos los códigos de cabecera tarea que estén listos o instanciados y que la versión esté en true. Si el tipo es "I" (inicio) devuelve los que tengan diagrama con evento de fin de mensajes y si el tipo es "F" (fin) devuelve los que tengan diagrama con evento de inicio de mensajes.	Tipo: "i" inicio, "f" fin e id cabecera actual.	Lista de cabecera tarea (procesos)	API/FiltroMensaje/5

Fuente: Autores.

### 1.14.37 FILTRO PROCESO DETALLE

Gestiona el filtro del detalle de los procesos administrativos registrados en la BD del sistema. A continuación, en el Cuadro 183 se presentan los métodos HTTP de este controlador.

**Cuadro 183.** Métodos HTTP de FiltroProcesoDetalleController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene el detalle del proceso.	Identity cabecera	Objeto cabecera tarea	API/FiltroProcesoDetalle/5

Fuente: Autores.

### 1.14.38 FILTRO PROCESO ID

Maneja el filtro de elementos específicos de los procesos y sus respectivas instancias registrados en la BD del sistema. A continuación, en el Cuadro 184Cuadro 180 se presentan los métodos HTTP de este controlador.

**Cuadro 184.** Métodos HTTP de FiltroProcesoIdController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene el Id del proceso y el Id de la instancia del trámite	Id del trámite	Diccionario (id del proceso, id de la instancia)	API/FiltroProcesoId/5

Fuente: Autores.

### 1.14.39 FILTRO REDISEÑO

Permite gestionar los procesos que admiten que el Administrador pueda aplicar rediseño cuando lo crea conveniente. A continuación, en el Cuadro 185 se presentan los métodos HTTP de este controlador.

**Cuadro 185.** Métodos HTTP de FiltroRedisenhoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Verifica si el proceso está listo para el rediseño.	Id cabecera	Estado: true si está listo, false si no lo está.	API/FiltroRedisenho/5

Fuente: Autores.

### 1.14.40 FILTRO REFERENCIA VD

Permite la ejecución de las actividades que contienen “vista de datos” durante la instancia de los procesos administrativos. A continuación, en el Cuadro 186 se presentan los métodos HTTP de este controlador.

**Cuadro 186.** Métodos HTTP de FiltroReferenciaVDController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Ejecuta una vista de datos y devuelve un objeto de referencia con su lista de elementos.	Id trámite	Objeto referencia formulario	API/FiltroReferenciaVD

Fuente: Autores.

#### 1.14.41 FILTRO TAREA

Permite gestionar el filtro del detalle de las tareas (integraciones, diagramación del proceso, diseño de los formularios, diseño de reportes, creación de vista de datos) que permiten el diseño o creación de un proceso. A continuación, en el Cuadro 187 se presentan los métodos HTTP de este controlador.

**Cuadro 187.** Métodos HTTP de FiltroTareaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene el detalle de la tarea.	Id de la tarea	Objeto tarea	API/FiltroTarea

Fuente: Autores.

#### 1.14.42 FILTRO ÚLTIMAS ALERTAS

Encargado de gestionar las notificaciones (de tipo alerta) más recientes enviadas por el sistema para que: los usuarios (DBA, Diseñador, Modelador) puedan llevar a cabo las tareas que tienen a su cargo, en el tiempo previamente establecido por el Administrador; y para que éste último pueda realizar la aprobación o rechazo de las tareas recibidas, llevar a cabo la asignación de recursos cuando el proceso esté listo para dicha etapa y en general, para que pueda controlar la evolución del proceso durante la creación o diseño del mismo. A continuación, en el Cuadro 188 se presentan los métodos HTTP de este controlador.

**Cuadro 188.** Métodos HTTP de FiltroÚltimasAlertasController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene las últimas 5 alertas.	Identity del usuario	Lista de mensaje tarea	API/FiltroÚltimasAlertas

Fuente: Autores.

### 1.14.43 HISTORIAL INSTANCIA

Permite registrar todas las acciones (ejecución, pausado, eliminación, reasignación de participantes, reinicio de tarea, reinicio completo de la instancia) que se presentan durante la ejecución de los procesos administrativos de la ESPAM MFL. Guarda en la BD del sistema datos importantes como: fecha, detalles u observación, trámite donde se presentó el cambio, entre otros, de dichos sucesos para controles o seguimientos posteriores. A continuación, en el Cuadro 189 se presentan los métodos HTTP de este controlador.

**Cuadro 189.** Métodos HTTP de HistorialInstanciaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene todo el historial de los cambios de estado de cada instancia.	Id instancia	Lista de historial instancia	API/HistorialInstancia/5
Post	Guarda el cambio de estado de una instancia.	Objeto historial instancia	-	API/HistorialInstancia

Fuente: Autores.

### 1.14.44 HISTORIAL TAREA

Permite registrar todos los cambios (creación, modificación, reasignación de responsable, cambio de fecha límite) que se presentan en las tareas, durante el diseño de un proceso administrativo. A continuación, en el Cuadro 190 se presentan los métodos HTTP de este controlador.

**Cuadro 190.** Métodos HTTP de HistorialTareaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los cambios realizados en la tarea.	Id tarea	Lista de historial tarea	API/HistorialTarea/5

Fuente: Autores.

### 1.14.45 HORARIO

Destinado a la gestión de los horarios laborales que tienen los trabajadores administrativos, facilitando la adaptación de los trámites de los procesos a la carga horaria establecida en el contrato del personal administrativo,

permitiendo que los plazos de entrega de dichas tareas sean realistas y accesibles a la disponibilidad profesional de los mismos. A continuación, en el Cuadro 191 se presentan los métodos HTTP de este controlador.

**Cuadro 191.** Métodos HTTP de HorarioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de todos los horarios administrativos	-	Lista de horarios	API/Horario
Get	Obtiene el detalle de un horario específico.	Id horario	Objeto horario	API/Horario/5

Fuente: Autores.

#### 1.14.46 INSTANCIA

Luego de que la etapa de diseño del proceso haya concluido con éxito, el usuario Operador encargado del inicio del mismo, podrá llevar a cabo la instancia del mismo. Durante la ejecución del proceso, el sistema permite que se lleven a cabo todas las actividades (eventos, tarea y compuertas) definidas previamente en la etapa inicial (diseño) del mismo y que el flujo se lleve a cabo de acuerdo a sus recursos asignados. A continuación, en el Cuadro 192 se presentan los métodos HTTP de este controlador.

**Cuadro 192.** Métodos HTTP de InstanciaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de las instancias disponibles en la BD del sistema.	-	Lista de instancias	API/Instancia
Get	Obtiene el detalle de una instancia en particular.	Id instancia	Objeto instancia	API/Instancia/5
Post	Guarda la instancia de un proceso	Objeto instancia	Lista de mensaje trámite	API/Instancia
Put	Actualiza (reinicia) la instancia de un proceso.	Objeto instancia	Lista de mensaje trámite	API/Instancia/5
Delete	Elimina por completo una instancia.	Id instancia	-	API/Instancia/5

Fuente: Autores.

#### 1.14.47 INTEGRACIÓN EXTERNA

Gestiona las tareas del tipo “integración externa” durante el diseño del proceso. Se realiza la consulta de información a sistemas externos al Workflow mediante

la ejecución de servicios web. A continuación, en el Cuadro 193 se presentan los métodos HTTP de este controlador.

**Cuadro 193.** Métodos HTTP de IntegraciónExternaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Get</b>	Obtiene los datos de la Integración Externa.	Id tarea	Objeto integración externa	API/IntegraciónExterna/5
<b>Post</b>	Guarda los elementos de la Integración Externa	Objeto integración externa	Estado: true, si es correcto y false si no es correcto.	API/IntegraciónExterna

Fuente: Autores.

#### 1.14.48 INTEGRACIÓN INTERNA

Cuando se requiere consultar información almacenada en la BD del sistema, el núcleo permite que se ejecuten las integraciones internas de datos, a fin de permitir que el Operador o Supervisor del proceso puedan utilizar dicha información para generar reportes que permitan contribuir en la toma de decisiones. A continuación, en el Cuadro 194 se presentan los métodos HTTP de este controlador.

**Cuadro 194.** Métodos HTTP de IntegraciónInternaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
<b>Get</b>	Consulta una integración interna, según el id tarea.	Id tarea	Objeto integración interna	API/IntegraciónInterna/5
<b>Post</b>	Recibe y guarda una integración interna.	Objeto integración interna	-	API/IntegraciónInterna
<b>Put</b>	Ejecución de la Integración Interna	Objeto integración interna	Objeto integración interna	API/IntegraciónInterna/5

Fuente: Autores.

#### 1.14.49 MENSAJE

Permite que el sistema envíe notificaciones (sencillas o de tipo alerta) a los usuarios encargados del diseño o creación de los procesos (Administrador, DBA, Modelador, Diseñador), contribuyendo a la realización a tiempo de las tareas de los mismos. Además, facilitan el acceso al Centro de Tareas del

sistema, donde se podrán visualizar y llevar a cabo todas las tareas asignadas a cada responsable, según el área de trabajo. A continuación, en el Cuadro 195 se presentan los métodos HTTP de este controlador.

**Cuadro 195.** Métodos HTTP de MensajeController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Retorna una lista de los mensajes correspondientes al id usuario enviado.	Identity del usuario, rol del usuario	Lista de mensaje tarea	API/Mensaje/5
Put	Modifica el estado de los mensajes a leídos	Id del mensaje	-	API/Mensaje/5

Fuente: Autores.

#### 1.14.50 MENSAJE TRÁMITE

Permite gestionar los mensajes enviados por el sistema a los usuarios Operadores durante las instancias de los procesos, a fin de que puedan completar sus tareas y a los Supervisores para que verifiquen la correcta evolución de esas instancias. En caso de errores, de igual forma se notifica al Supervisor para que pueda tomar las medidas correctivas necesarias. A continuación, en el Cuadro 196 se presentan los métodos HTTP de este controlador.

**Cuadro 196.** Métodos HTTP de MensajeTramiteController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de mensajes tramite de un usuario con un rol específico	Identity del usuario, rol del usuario	Lista de mensaje trámite	API/MensajeTrámite/5

Fuente: Autores.

#### 1.14.51 OPERADOR PROCESO

Permite gestionar los usuarios Operadores responsables de instanciar o iniciar los procesos administrativos. A continuación, en el Cuadro 197 se presentan los métodos HTTP de este controlador.

**Cuadro 197.** Métodos HTTP de FiltroMensajeController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los procesos que inicia el usuario, por identity o cargo.	Identity del usuario	Lista de cabecera tarea	API/OperadorProceso/5

Fuente: Autores.

### 1.14.52 PROCESO

Encargado de gestionar aspectos importantes relacionados a los procesos administrativos del sistema Workflow. A continuación, en el Cuadro 198 se presentan los métodos HTTP de este controlador.

**Cuadro 198.** Métodos HTTP de ProcesoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene el modelo de la versión activa correspondiente al código del proceso enviado	Código único del proceso	XML del proceso.	API/Proceso/5
Post	Establece una nueva fecha límite para el proceso, en modo rediseño.	objeto cabecera	True si la fecha se establece correctamente, caso contrario false.	API/Proceso
Put	Establece una nueva fecha límite para una tarea, en modo rediseño.	Objeto tarea	True si la fecha se establece correctamente, caso contrario false.	API/Proceso/5

Fuente: Autores.

### 1.14.53 REFERENCIA FORMULARIO

Gestiona las referencias de los formularios (documentos HTML) realizadas por el usuario Diseñador. A continuación, en el Cuadro 199 se presentan los métodos HTTP de este controlador.

**Cuadro 199.** Métodos HTTP de ReferenciaFormularioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene el objeto de referencia de una tarea.	Id tarea	Objeto referencia formulario	API/ReferenciaFormulario/5
Post	Guarda la referencia a un formulario.	Objeto referencia formulario	-	API/ReferenciaFormulario
Delete	Elimina una referencia a un formulario con sus elementos	Id tarea referenciada	-	API/ReferenciaFormulario/5

Fuente: Autores.

### 1.14.54 REFERENCIA INTEGRACIÓN

Gestiona las referencias a integraciones. A continuación, en el Cuadro 200 se presentan los métodos HTTP de este controlador.

**Cuadro 200.** Métodos HTTP de ReferencialIntegraciónController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los datos de una referencia de integración.	id de la tarea	Objeto referencia integración	API/ReferencialIntegración/5
Post	Guarda una referencia de integración.	Objeto referencia integración	-	API/ReferencialIntegración

Fuente: Autores.

### 1.14.55 REGLA

Controla todo lo relacionado a las reglas establecidas durante la asignación de recursos en las compuertas para la toma de decisiones del flujo del proceso (diagrama) durante la instancia del mismo. A continuación, en el Cuadro 201 se presentan los métodos HTTP de este controlador.

**Cuadro 201.** Métodos HTTP de ReglaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los datos necesario para la creación de la regla	Id de la versión del elemento (compuerta) a la que se asignará la regla.	Objeto regla	API/Regla/5
Post	Guarda una regla correspondiente a una compuerta.	Objeto regla	-	API/Regla

Fuente: Autores.

### 1.14.56 REPORTE

Gestiona la repostería del sistema, misma que no ha sido definida de forma estática, sino que permite que los usuarios adecuados (Administrador y Supervisor) puedan generar los documentos necesarios sobre el comportamiento de los procesos y usuarios que tienen a su cargo. A continuación, en el Cuadro 202 se presentan los métodos HTTP de este controlador.

**Cuadro 202.** Métodos HTTP de ReporteController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de reportes.	-	Lista de reportes	API/Reporte
Post	Guarda un nuevo reporte.	Objeto reporte	-	API/Reporte

Fuente: Autores.

### 1.14.57 RESPONSABLE

Permite gestionar todo lo referente a los usuarios Operadores, encargados de llevar a cabo las tareas (trámites) durante la ejecución de los procesos administrativos del sistema. A continuación, en el Cuadro 203 se presentan los métodos HTTP de este controlador.

**Cuadro 203.** Métodos HTTP de ResponsableController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los trámites pendientes de un responsable específico.	Lista de procesos para consultar los trámites, usuario responsable para buscar	Lista de trámites	API/Responsable/5
Put	Actualiza al operador responsable de una lista de trámites.	lista de trámites a modificar	Lista de mensajes trámite	API/Responsable/5

Fuente: Autores.

### 1.14.58 ROLES

Permite gestionar los roles de los usuarios almacenados en la BD del sistema. A continuación, en el Cuadro 204 se presentan los métodos HTTP de este controlador.

**Cuadro 204.** Métodos HTTP de RolesController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de usuarios según el rol enviado.	Rol	Lista de usuarios	API/Roles

Fuente: Autores.

### 1.14.59 SUPERVISIÓN

Controlador con acciones necesarias para la supervisión de las tareas de definición de un proceso. A continuación, en el Cuadro 205 se presentan los métodos HTTP de este controlador.

**Cuadro 205.** Métodos HTTP de SupervisiónController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Post	Actualiza la fecha límite de una tarea.	Objeto cabecera tarea	Lista de mensaje tarea	API/Supervisión
Put	Actualiza la fecha límite y el responsable de una tarea	Objeto historial tarea	Lista de mensaje tarea	API/Supervisión/5

Fuente: Autores.

### 1.14.60 TAREA

Gestiona los datos de las tareas asignadas a los usuarios DBA, Diseñador y Modelador durante la creación o diseño de los procesos. A continuación, en el Cuadro 206 se presentan los métodos HTTP de este controlador.

**Cuadro 206.** Métodos HTTP de TareaController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene la información completa de una tarea específica.	Id tarea	Lista de mensaje tarea	API/Tarea/5
Post	Crea una nueva tarea para un proceso creado	Objeto tarea	Lista de mensaje tarea	API/Tarea
Put	Guarda tareas para modificar la versión de un proceso	Objeto tarea	Lista de mensaje tarea	API/Tarea/5
Delete	Elimina una tarea.	Id tarea	Lista de mensaje tarea	API/Tarea/5

Fuente: Autores.

### 1.14.61 TIPO DATO

Gestiona los tipos de datos en la BD del sistema, permitiendo establecer tipos de datos personalizados, necesarios para los elementos de los formularios HTML, las integraciones internas y externas, entre otras entidades. A continuación, en el Cuadro 207 se presentan los métodos HTTP de este controlador.

**Cuadro 207.** Métodos HTTP de TipoDatoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de los tipos de datos guardados en la BD.	-	Lista de tipos de datos.	API/TipoDato

Fuente: Autores.

### 1.14.62 TRÁMITE

Gestiona las tareas (trámites) necesarios para que un flujo de proceso pueda ejecutarse. Cada elemento XML presente en el diagrama de flujo (eventos, compuertas y actividades) representa un trámite con responsable, fecha de inicio, fecha de finalización, fecha límite, entre otros atributos. A continuación, en el Cuadro 208 se presentan los métodos HTTP de este controlador.

**Cuadro 208.** Métodos HTTP de TrámiteController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene los datos del trámite que se va a realizar (F, RF, IVD, VDF, IR)	Id trámite	Objeto trámite	API/Trámite/5
Put	Actualiza fecha límite de un trámite.	Objeto trámite	Lista de mensaje trámite	API/Trámite/5

Fuente: Autores.

### 1.14.63 USUARIO

Gestiona todos lo relacionado a los usuarios del sistema. A continuación, en el Cuadro 209 se presentan los métodos HTTP de este controlador.

**Cuadro 209.** Métodos HTTP de UsuarioController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene una lista de usuarios guardados en la Base de Datos.	-	Lista de usuarios	API/Usuario
Get	Obtiene una lista de usuarios según el identity enviado.	Identity del usuario	Objeto usuario	API/Usuario/5
Post	Guarda un nuevo usuario.	Objeto usuario	-	API/Usuario
Put	Actualiza el registro de un Usuario.	Objeto usuario	-	API/Usuario/5
Delete	Elimina un usuario de la Base de Datos.	Id usuario	-	API/Usuario/5

Fuente: Autores.

### 1.14.64 VERSIÓN

Un proceso administrativo puede presentar cambios a través del tiempo, por lo cual el Workflow se adapta a dichos cambios y permite que se creen versiones del mismo, pudiendo añadir o quitar elementos del diagrama original. A continuación, en el Cuadro 210 se presentan los métodos HTTP de este controlador.

**Cuadro 210.** Métodos HTTP de VersiónController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Get	Obtiene todas las versiones de un proceso.	Id proceso	Lista de versiones	API/Versión/5
Post	Guarda una nueva versión de un proceso existente.	Objeto versión	-	API/Versión

Fuente: Autores.

### 1.14.65 VERSIÓN ELEMENTO

Permite gestionar los elementos que forman el diagrama de flujo de los procesos administrativo. Pueden ser eventos (de inicio o fin), compuertas (divergentes o convergentes) o actividades (automáticas, manuales o de usuario). Todos tienen su propia definición y composición, siguiendo lo establecido en el estándar BPMN 2.0 (Anexo 4). A continuación, en el Cuadro 211 se presentan los métodos HTTP de este controlador.

**Cuadro 211.** Métodos HTTP de VersiónElementoController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Put	Agrega ciertos recursos a los objetos del diagrama de flujo de proceso.	Objeto versión elemento	-	API/VersiónElemento/5

Fuente: Autores.

### 1.14.66 WEB SERVICE

Controlador que valida la url enviada por el cliente. A continuación, en el Cuadro 212 se presentan los métodos HTTP de este controlador.

**Cuadro 212.** Métodos HTTP de WebServiceController.

Método HTTP	Descripción	Parámetro(s) que recibe	Parámetro que retorna	Ruta o URL
Put	Establece si el servicio web funciona de manera correcta con los parámetros enviados	Objeto integración externa	Objeto integración externa	API/IntegraciónExterna/5
Post	Obtiene los datos de un servicio web	Objeto integración externa	Objeto integración externa	API/IntegraciónExterna/5

Fuente: Autores.

## METADATA

### 1.15 DATABASE FIRST

A fin de implementar Entity Framework en la aplicación, se utiliza uno de los primeros métodos disponibles (y quizás el más sencillo), denominado “BD primero”. Y es que tal y como su nombre lo indica, en primer lugar se crea la BD con sus tablas (y otras estructuras) para luego incorporarlas a la aplicación. Para lo cual, se utiliza el SGBD SQL Server 2008 donde se crearon los diferentes esquemas, entidades y diagramas.

### 1.16 ESQUEMAS

En el Cuadro 213 se describen los diferentes esquemas de la BD, es importante mencionar que existen tablas que no pertenecen a ningún esquema, por lo cual están ubicadas de forma independiente (.dbo).

**Cuadro 213.** Descripción de los esquemas de la BD.

Esquema	Descripción
dbo	Esquema general (por defecto).
usuario	Encargado de la gestión de los usuarios para la ejecución de los procesos y también para los responsables de crear nuevos procesos.
diseñoproceso	Encargado de contener las tablas para la creación, actualización (versionamiento) y eliminación (borrado lógico) de los procesos y sus respectivas versiones, y las actividades de cada proceso.
instanciaproceso	Destinado al levantamiento o ejecución de los procesos, luego de que han sido almacenados en la BD.
tarea	Establecido para gestionar las responsabilidades de los usuarios encargadas por el Administrador del proceso, cada vez que se requiera automatizar un nuevo proceso.

Elaboración: Autores

## 1.17 DIAGRAMAS ENTIDAD-RELACIÓN

A continuación, en las figuras 21, 22, 23 y 24 se presentan los diagramas que componen las entidades de cada esquema mencionado anteriormente (excepto el esquema general).

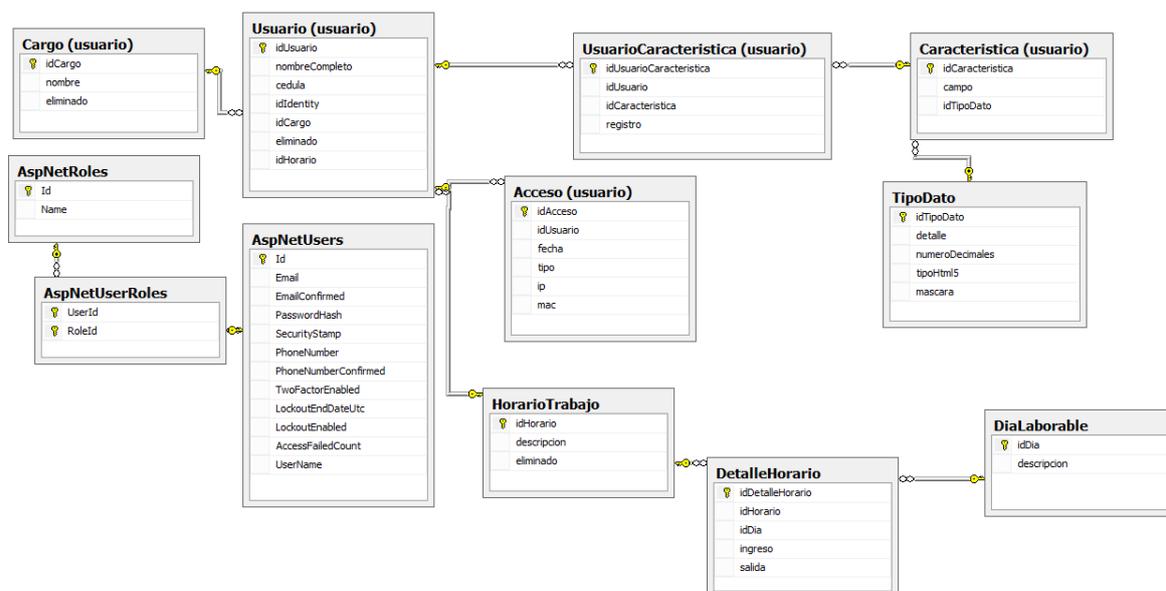


Figura 48. Diagrama del esquema usuario.  
Elaboración: Autores

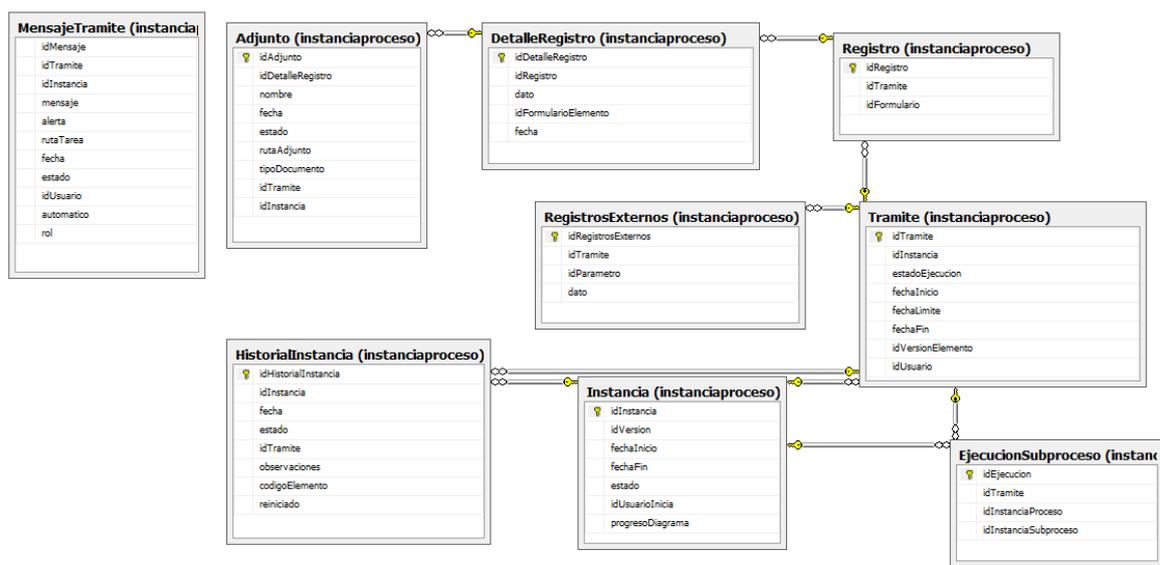


Figura 49. Diagrama del esquema instanciaproceso.  
Elaboración: Autores

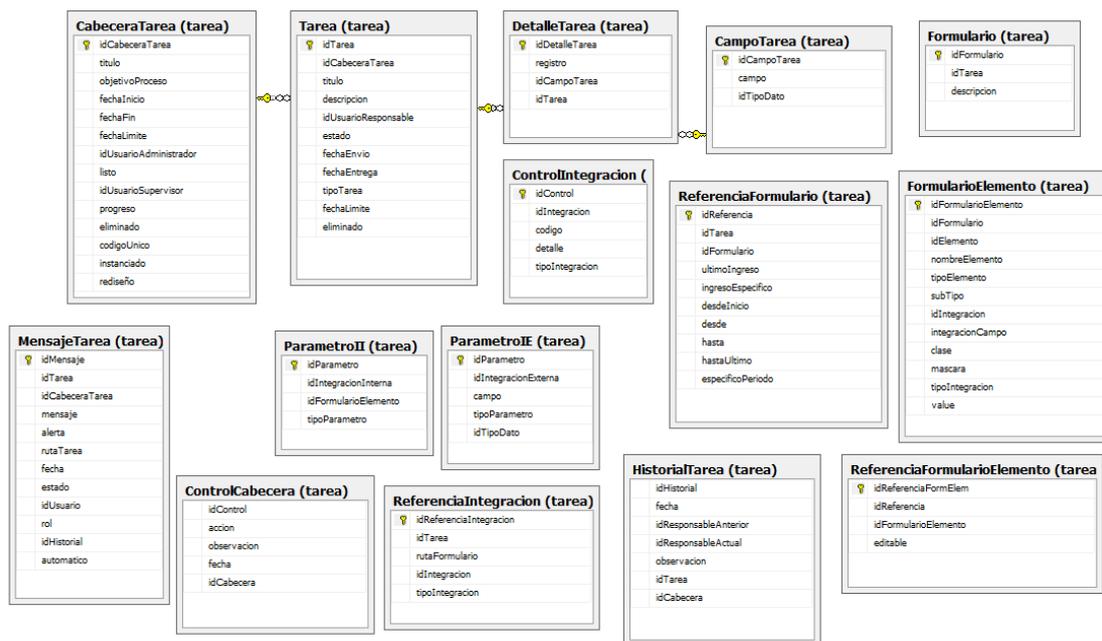


Figura 50. Diagrama del esquema tarea.

Elaboración: Autores

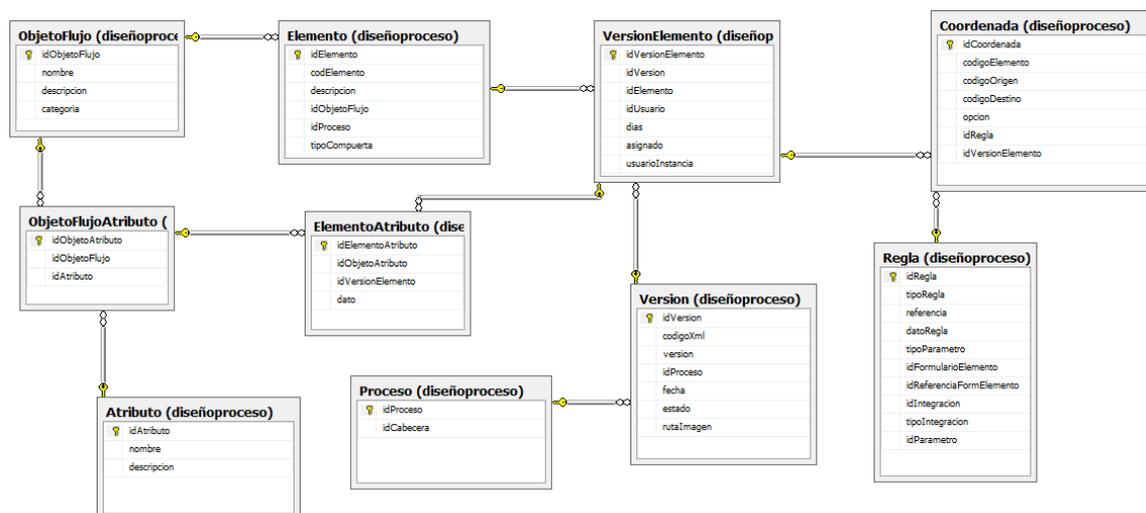


Figura 51. Diagrama del esquema diseño proceso.

Elaboración: Autores

Para conectar la BD con la aplicación se agrega el archivo de Entity Framework a la API web, mediante un nuevo elemento de tipo “ADO.NET Entity Data Model”, mismo que se denomina “Conexión”. La conexión se “generar desde la base de datos”. A continuación, se ubica el servidor local y se selecciona la BD “Motor\_Workflow”, finalmente se añaden todos los procedimientos almacenados previamente codificados en la BD. Finalmente se obtiene el

archivo Web.config con el string de conexión “WorkflowEntities” configurado como se muestra en la Figura 52.

```
<connectionStrings>
  <add name="DefaultConnection"
    connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-motorWorkflow-20170110051322.mdf;Initial
    Catalog=aspnet-motorWorkflow-20170110051322;Integrated Security=True" providerName="System.Data.SqlClient" />
  <add name="WorkflowEntities" connectionString="metadata=res://*/Conexion.csdl|res://*/Conexion.ssdl|res://*/
    Model1.msdl;provider=System.Data.SqlClient;provider connection string=&quot;data source=ADMIN\SQLEXPRESSDB;initial catalog=Motor_Workflow;
    integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;" providerName="System.Data.EntityClient" />
</connectionStrings>
```

**Figura 52.** ConnectionStrings.  
Elaboración: Autores

## 1.18 DESCRIPCIÓN DE TABLAS Y DATOS

En el Cuadro 214 se presentan todas las tablas correspondientes a la BD Motor\_Workflow, incluye una breve descripción y se especifica a qué esquema pertenece cada tabla.

**Cuadro 214.** Tablas pertenecientes a la BD Motor\_Workflow

Esquema	Tabla	Descripción
dbo	__MigrationHistory	Guarda las migraciones de asp.
dbo	AspNetRoles	Guarda los roles a utilizarse en el sistema
dbo	AspNetUserLogins	Guarda los accesos al sistema
dbo	AspNetUserRoles	Tabla usada para asignar roles a los diferentes usuarios
dbo	AspNetUsers	Guarda los datos de los usuarios
dbo	ControlCabecera	Guarda las acciones (CRUD) realizadas en una cabecera
dbo	DetalleHorario	Guarda el detalle de los horarios
dbo	DiaLaborable	Guarda el detalle de los día laborables
dbo	HorarioTrabajo	Guarda los horarios de trabajo
dbo	Reporte	Guarda los reportes generados de la versión de un proceso
dbo	TipoDato	Guarda los diferentes tipos de datos a utilizar
diseñoproceso	Atributo	Guarda los atributos de los elementos del BPMN2.0
diseñoproceso	Coordenada	Tabla que guarda las rutas que sigue el proceso
diseñoproceso	Elemento	Guarda cada uno de los elementos de un proceso
diseñoproceso	ElementoAtributo	Tabla intermedia que relaciona los elementos de un proceso con sus atributos
diseñoproceso	ObjetoFlujo	Guarda cada uno de los elementos del BPMN2.0
diseñoproceso	ObjetoFlujoAtributo	Tabla intermedia que relaciona los elementos del BPMN2.0 con sus atributos
diseñoproceso	Proceso	Guarda una cabecera para cada proceso creado
diseñoproceso	Regla	Guarda las reglas que se usaran para la toma de decisiones en una compuerta
diseñoproceso	Version	Guarda las versiones que puede tener un proceso
diseñoproceso	VersionElemento	Tabla que relaciona los elementos que pertenecen a la versión de un proceso
instanciaproceso	Adjunto	Guarda el historial de los documentos adjuntos de una actividad
instanciaproceso	DetalleRegistro	Guarda los detalles de los registros que se realicen en un trámite
instanciaproceso	EjecucionSubproceso	Guarda los procesos que actúan como subprocesos.

<b>instanciaproceso</b>	HistorialInstancia	Guarda el historial de las instancias de un proceso
<b>instanciaproceso</b>	Instancia	Guarda un registro cuando un proceso es instanciado
<b>instanciaproceso</b>	MensajeTramite	Guarda los mensajes a los operadores de los trámites.
<b>instanciaproceso</b>	Registro	Guarda una cabecera para los registros realizados en un trámite
<b>instanciaproceso</b>	RegistrosExternos	Guarda los datos de entrada para la ejecución de una Integración Externa
<b>instanciaproceso</b>	Tramite	Guarda los trámites que se deben realizar para cumplir con el proceso
<b>tarea</b>	CabeceraTarea	Se utiliza para guardar un encabezado de la lista de tareas de un proceso
<b>tarea</b>	CampoTarea	Guarda los campos que tendrá una tarea
<b>tarea</b>	ControlCabecera	Guarda las acciones (CRUD) realizadas en una cabecera
<b>tarea</b>	ControllIntegracion	Guarda los errores que se presenten al ejecutar una integración
<b>tarea</b>	DetalleTarea	Guarda el desarrollo de la tarea
<b>tarea</b>	Formulario	Guarda los formularios creados
<b>tarea</b>	FormularioElemento	Guarda los elementos de un formulario
<b>tarea</b>	HistorialTarea	Guarda un registro cuando existen modificaciones de responsable de una tarea
<b>tarea</b>	IntegracionExterna	Guarda los datos necesarios para realizar la Integración Externa
<b>tarea</b>	IntegracionInterna	Guarda los datos necesarios para realizar una Integración Interna
<b>tarea</b>	MensajeTarea	Guarda un registro con un mensaje para el usuario encargado de realizar una actividad
<b>tarea</b>	ParametroIE	Guarda los parámetros de una Integración Externa
<b>tarea</b>	ParametroII	Guarda los parámetros de una Integración Interna
<b>tarea</b>	ReferenciaFormulario	Guarda las referencias hechas a un formulario
<b>tarea</b>	ReferenciaFormularioElemento	Guarda la referencia a los elementos de un formulario
<b>tarea</b>	ReferencialIntegracion	Guarda una referencia a una integración
<b>tarea</b>	Tarea	Guarda las tareas que se deben realizar para la creación de un proceso
<b>usuario</b>	Acceso	Se utiliza para guardar los accesos de los usuarios al sistema
<b>usuario</b>	Caracteristica	Guarda las características de los usuarios
<b>usuario</b>	Cargo	Guarda los cargos que se utilizan en el sistema
<b>usuario</b>	Usuario	Tabla que guarda la información principal del usuario
<b>usuario</b>	UsuarioCaracteristica	Tabla intermedia que guarda las diversas características de un usuario

Elaboración: Autores

En el Cuadro 215 se muestra la descripción y características de cada columna de las tablas pertenecientes al esquema general (.dbo). Además, se detalla si el campo admite nulos y si es o no una clave primaria (PK).

**Cuadro 215.** Columnas correspondientes a las tablas del esquema general (.dbo).

Tabla	Columna	Tipo de dato	Nulos	PK	Descripción
<b>__MigrationHistory</b>	MigrationId	NVarChar (150)	N	Y	<b>Clave Principal de la tabla</b>
<b>__MigrationHistory</b>	ContextKey	NVarChar (300)	N	Y	<b>Clave del contexto de la migración</b>
<b>__MigrationHistory</b>	Model	VarBinaryMax	N	N	<b>Modelo migrado.</b>

<b>__MigrationHistory</b>	ProductVersion	NVarChar (32)	N	N	<b>Versión del producto.</b>
<b>AspNetRoles</b>	Id	NVarChar (128)	N	Y	<b>Clave principal de la tabla</b>
<b>AspNetRoles</b>	Name	NVarChar (256)	N	N	<b>Nombre del rol</b>
<b>AspNetUserLogins</b>	LoginProvider	NVarChar (128)	N	Y	<b>Clave principal de la tabla</b>
<b>AspNetUserLogins</b>	ProviderKey	NVarChar (128)	N	Y	<b>Token de acceso</b>
<b>AspNetUserLogins</b>	UserId	NVarChar (128)	N	Y	<b>Id del usuario</b>
<b>AspNetUserRoles</b>	UserId	NVarChar (128)	N	Y	<b>Clave principal de la tabla</b>
<b>AspNetUserRoles</b>	RoleId	NVarChar (128)	N	Y	<b>Id del Rol</b>
<b>AspNetUsers</b>	Id	NVarChar (128)	N	Y	<b>Clave principal de la tabla</b>
<b>AspNetUsers</b>	Email	NVarChar (256)	Y	N	<b>Correo del Usuario</b>
<b>AspNetUsers</b>	EmailConfirmed	Bit	N	N	<b>Indica si el usuario ha confirmado el correo</b>
<b>AspNetUsers</b>	PasswordHash	NVarCharMax	Y	N	<b>Contraseña encriptada</b>
<b>AspNetUsers</b>	SecurityStamp	NVarCharMax	Y	N	<b>Sello de seguridad para el inicio de sesión</b>
<b>AspNetUsers</b>	PhoneNumber	NVarCharMax	Y	N	<b>Número de teléfono</b>
<b>AspNetUsers</b>	PhoneNumberConfirmed	Bit	N	N	<b>Indica si el usuario ha confirmado su número de teléfono</b>
<b>AspNetUsers</b>	TwoFactorEnabled	Bit	N	N	<b>Indica si la autenticación se puede hacer de dos formas</b>
<b>AspNetUsers</b>	LockoutEndDateUtc	DateTime	Y	N	<b>Fecha en la que el usuario ha sido bloqueado</b>
<b>AspNetUsers</b>	LockoutEnabled	Bit	N	N	<b>Indica si el bloqueo de usuario está habilitado</b>
<b>AspNetUsers</b>	AccessFailedCount	Int	N	N	<b>Contador de accesos fallidos</b>
<b>AspNetUsers</b>	UserName	NVarChar (256)	N	N	<b>Nombre de usuario</b>
<b>ControlCabecera</b>	idControl	Int	N	Y	<b>Clave principal de la tabla</b>
<b>ControlCabecera</b>	accion	VarChar (50)	N	N	<b>Especifica el tipo de sentencia sql se realizó</b>
<b>ControlCabecera</b>	observacion	VarCharMax	N	N	<b>Indica el motivo del registro</b>
<b>ControlCabecera</b>	fecha	DateTime	N	N	<b>Fecha en la que se crea el registro</b>
<b>ControlCabecera</b>	idCabecera	Int	N	N	<b>Id de la cabecera a la que pertenece el registro</b>
<b>DetalleHorario</b>	idDetalleHorario	Int	N	Y	<b>Clave principal de la tabla</b>
<b>DetalleHorario</b>	idHorario	Int	N	N	<b>Id del horario al que corresponde el detalle</b>
<b>DetalleHorario</b>	idDia	Int	N	N	<b>Id del día al que corresponde el registro</b>
<b>DetalleHorario</b>	ingreso	Time	N	N	<b>Hora de ingreso</b>
<b>DetalleHorario</b>	salida	Time	N	N	<b>Hora de salida</b>
<b>DiaLaborable</b>	idDia	Int	N	Y	<b>Clave principal de la tabla</b>
<b>DiaLaborable</b>	descripcion	VarChar (20)	N	N	<b>Nombre del día</b>
<b>DiaLaborable</b>	descripcion2	VarChar (20)	N	N	
<b>HorarioTrabajo</b>	idHorario	Int	N	Y	<b>Clave principal de la tabla</b>
<b>HorarioTrabajo</b>	descripcion	VarChar (50)	N	N	<b>Descripción del horario</b>

<b>HorarioTrabajo</b>	eliminado	Bit	N	N	Indica si el horario está o no eliminando
<b>Reporte</b>	idReporte	Int	N	Y	Clave principal de la tabla
<b>Reporte</b>	idUsuarioidentity	Int	N	N	Usuario que realiza el reporte
<b>Reporte</b>	fecha	DateTime	N	N	Fecha de creación del reporte
<b>Reporte</b>	ruta	VarCharMax	N	N	Ruta para la generación del pdf
<b>Reporte</b>	descripcion	VarCharMax	N	N	Descripción del reporte
<b>Reporte</b>	tipo	VarChar (100)	N	N	Establece el tipo de reporte que se crea
<b>TipoDato</b>	idTipoDato	Int	N	Y	Clave principal de la tabla
<b>TipoDato</b>	detalle	VarChar (100)	N	N	Nombre del tipo de dato
<b>TipoDato</b>	numeroDecimales	VarChar (50)	Y	N	Cantidad de decimales
<b>TipoDato</b>	tipoHtml5	VarChar (20)	Y	N	Tipo de dato HTML
<b>TipoDato</b>	mascara	VarChar (50)	Y	N	Establece una máscara para el tipo de dato

Elaboración: Autores

Por su parte, el Cuadro 216 describe los campos de las tablas que pertenecen al esquema “diseñoproceso”.

**Cuadro 216.** Columnas correspondientes a las tablas del esquema "diseñoproceso".

Tabla	Atributo	Tipo de dato	Nulo	PK	Descripción
<b>Atributo</b>	idAtributo	Int	N	Y	Clave principal de la tabla
<b>Atributo</b>	nombre	VarChar (1000)	N	N	Nombre del atributo
<b>Atributo</b>	descripción	VarChar (500)	Y	N	Breve descripción del atributo
<b>Coordenada</b>	idCoordenada	Int	N	Y	Clave principal de la tabla
<b>Coordenada</b>	codigoElemento	NVarChar (50)	N	N	Código del elemento
<b>Coordenada</b>	codigoOrigen	NVarChar (50)	Y	N	Código que indica el elemento de origen
<b>Coordenada</b>	codigoDestino	NVarChar (50)	Y	N	Código que indica el código de destino
<b>Coordenada</b>	opcion	VarChar (50)	Y	N	Campo usado en los elementos de compuertas, que guarda la opción de cada camino o ruta
<b>Coordenada</b>	idRegla	Int	Y	N	Id de la Regla que se usa en la compuerta
<b>Coordenada</b>	idVersionElemento	Int	Y	N	Id de la versión del elemento al que pertenece la coordenada
<b>Elemento</b>	idElemento	Int	N	Y	Clave principal de la tabla
<b>Elemento</b>	codElemento	NVarChar (50)	N	N	Código del elemento
<b>Elemento</b>	descripción	VarCharMax	Y	N	Breve descripción de lo que se realiza en ese elemento
<b>Elemento</b>	idObjetoFlujo	VarChar (50)	N	N	Código del tipo de objeto
<b>Elemento</b>	idProceso	Int	Y	N	Id del Proceso al que

					<b>pertenece el elemento</b>
<b>Elemento</b>	tipoCompuerta	NVarChar (2)	Y	N	<b>Define si la compuerta es Divergente o Convergente</b>
<b>ElementoAtributo</b>	idElementoAtributo	Int	N	Y	<b>Clave principal de la tabla</b>
<b>ElementoAtributo</b>	idObjetoAtributo	Int	N	N	<b>Id del Atributo</b>
<b>ElementoAtributo</b>	idVersionElemento	Int	N	N	<b>Id de la versión del Elemento</b>
<b>ElementoAtributo</b>	dato	VarChar (1000)	Y	N	<b>Registro asociado al atributo</b>
<b>ObjetoFlujo</b>	idObjetoFlujo	VarChar (50)	N	Y	<b>Clave principal de la tabla</b>
<b>ObjetoFlujo</b>	nombre	VarChar (100)	N	N	<b>Nombre del objeto</b>
<b>ObjetoFlujo</b>	descripcion	VarCharMax	N	N	<b>Breve descripción del objeto</b>
<b>ObjetoFlujo</b>	categoria	VarChar (50)	Y	N	<b>Categoría a la que pertenece el objeto</b>
<b>ObjetoFlujoAtributo</b>	idObjetoAtributo	Int	N	Y	<b>Clave principal de la tabla</b>
<b>ObjetoFlujoAtributo</b>	idObjetoFlujo	VarChar (50)	N	N	<b>Id del objeto</b>
<b>ObjetoFlujoAtributo</b>	idAtributo	Int	N	N	<b>Id del atributo</b>
<b>Proceso</b>	idProceso	Int	N	Y	<b>Clave principal de la tabla</b>
<b>Proceso</b>	idCabecera	Int	N	N	<b>Id de la cabecera que corresponde al proceso</b>
<b>Regla</b>	idRegla	Int	N	Y	<b>Clave principal de la tabla</b>
<b>Regla</b>	tipoRegla	VarChar (10)	Y	N	<b>Tipo de regla que se crea</b>
<b>Regla</b>	referencia	VarChar (10)	Y	N	<b>Dato de referencia</b>
<b>Regla</b>	datoRegla	VarChar (50)	Y	N	<b>Dato para ejecutar la regla</b>
<b>Regla</b>	tipoParametro	VarChar (20)	Y	N	<b>Tipo de parámetro que se usa para la ejecución de la regla</b>
<b>Regla</b>	idFormularioElemento	Int	Y	N	<b>Id del Formulario Elemento que se usa en la regla</b>
<b>Regla</b>	idReferenciaFormElemento	Int	Y	N	<b>Id de Referencia Formulario que se usa en la regla</b>
<b>Regla</b>	idIntegracion	Int	Y	N	<b>Id de la integración que se usa en la regla</b>
<b>Regla</b>	tipoIntegracion	VarChar (20)	Y	N	<b>Tipo de integración usado</b>
<b>Regla</b>	idParametro	Int	Y	N	<b>Id del parametro utilizado</b>
<b>Version</b>	idVersion	Int	N	Y	<b>Clave principal de la tabla</b>
<b>Version</b>	codigoXml	VarCharMax	N	N	<b>XML del diagrama de la versión</b>
<b>Version</b>	version	Int	N	N	<b>Número de la versión</b>
<b>Version</b>	idProceso	Int	N	N	<b>Proceso al que pertenece la versión</b>
<b>Version</b>	fecha	Date	N	N	<b>Fecha de creación de la versión</b>
<b>Version</b>	estado	Bit	N	N	<b>Estado de la versión</b>
<b>Version</b>	rutalmagen	VarCharMax	Y	N	<b>Ruta de la imagen del diagrama</b>
<b>VersionElemento</b>	idVersionElemento	Int	N	Y	<b>Clave principal de la tabla</b>
<b>VersionElemento</b>	idVersion	Int	N	N	<b>Versión a la que pertenece un elemento</b>
<b>VersionElemento</b>	idElemento	Int	N	N	<b>Elemento que interviene en el proceso</b>
<b>VersionElemento</b>	idUsuario	Int	Y	N	<b>Usuario responsable de la actividad</b>

VersionElemento	días	Int	Y	N	Tiempo de ejecución de la actividad
VersionElemento	asignado	Bit	Y	N	Establece si la versión del elemento tiene recursos asignados
VersionElemento	usuariInstancia	Bit	Y	N	Establece si el elemento es manejado por e

Elaboración: Autores

El Cuadro 217 describe los campos de las tablas que pertenecen al esquema "instanciaproceso".

**Cuadro 217.** Columnas correspondientes a las tablas del esquema "instanciaproceso".

Tabla	Atributo	Tipo de dato	Nulo	PK	Descripción
Adjunto	idAdjunto	Int	N	Y	Clave principal de la tabla
Adjunto	idDetalleRegistro	Int	Y	N	Id del adjunto al que pertenece el histórico
Adjunto	nombre	VarChar (1000)	N	N	Nombre del archivo
Adjunto	fecha	DateTime	N	N	Fecha de modificación del archivo
Adjunto	estado	NVarChar (2)	N	N	Estado actual del adjunto
Adjunto	rutaAdjunto	VarCharMax	N	N	Ruta donde se guarda el archivo
Adjunto	tipoDocumento	VarChar (50)	N	N	Extensión del archivo
Adjunto	idTramite	Int	Y	N	Id del trámite que usa el archivo
Adjunto	idInstancia	Int	Y	N	Id de la Instancia en la que se sube el archivo
DetalleRegistro	idDetalleRegistro	Int	N	Y	Clave principal de la tabla
DetalleRegistro	idRegistro	Int	N	N	Registro al que pertenecen los detalles
DetalleRegistro	dato	VarCharMax	N	N	Respuesta del campo
DetalleRegistro	idFormularioElemento	Int	N	N	Id del formulario Elemento (campo) que se guarda
DetalleRegistro	fecha	DateTime	N	N	Fecha de creación del registro
EjecucionSubproceso	idEjecucion	Int	N	Y	Clave principal de la tabla
EjecucionSubproceso	idTramite	Int	N	N	Trámite donde se ejecuta el subproceso.
EjecucionSubproceso	idInstanciaProceso	Int	N	N	Id de la instancia del proceso principal.
EjecucionSubproceso	idInstanciaSubproceso	Int	N	N	Id de la instancia del subproceso.
HistorialInstancia	idHistorialInstancia	Int	N	Y	Clave principal de la tabla
HistorialInstancia	idInstancia	Int	N	N	Id de la Instancia a la que pertenece el historial
HistorialInstancia	fecha	DateTime	N	N	Fecha en la que la instancia sufre un cambio de estado
HistorialInstancia	estado	NChar (1)	N	N	Guarda el tipo de estado que tiene la instancia

HistorialInstancia	idTramite	Int	Y	N	Trámite en el cuál se registra el cambio de estado
HistorialInstancia	observaciones	VarCharMax	Y	N	Observaciones de las acciones ocurridas sobre la instancia
HistorialInstancia	codigoElemento	VarCharMax	Y	N	Código único del elemento XML donde ocurre la instancia
HistorialInstancia	reiniciado	Bit	Y	N	Establece si registra el reinicio de la instancia.
Instancia	idInstancia	Int	N	Y	Clave principal de la tabla
Instancia	idVersion	Int	N	N	Versión del proceso a la que pertenece la instancia
Instancia	fechaInicio	DateTime	N	N	Fecha de inicio de la instancia
Instancia	fechaFin	DateTime	Y	N	Fecha de fin de la instancia
Instancia	estado	NChar (1)	N	N	Estado en el cuál se encuentra la instancia
Instancia	idUsuarioInicia	Int	Y	N	Usuario que inicia la instancia
Instancia	progresoDiagrama	VarCharMax	Y	N	Progreso de la instancia (en diagrama)
MensajeTramite	idMensaje	Int	N	N	Clave principal de la tabla
MensajeTramite	idTramite	Int	Y	N	Id del trámite al que pertenece el mensaje
MensajeTramite	idInstancia	Int	Y	N	Id de la instancia al que pertenece el mensaje
MensajeTramite	mensaje	VarCharMax	N	N	Mensaje del trámite
MensajeTramite	alerta	Bit	N	N	Establece si el mensaje es una alerta
MensajeTramite	rutaTarea	VarCharMax	Y	N	Ruta de la tarea a realizar
MensajeTramite	fecha	DateTime	N	N	Fecha de creación del mensaje
MensajeTramite	estado	NVarChar (2)	N	N	Estado del mensaje
MensajeTramite	idUsuario	Int	N	N	Id del Usuario al que pertenece el mensaje
MensajeTramite	automatico	Bit	N	N	Establece si el mensaje es automático
MensajeTramite	rol	NVarChar (20)	N	N	Rol del usuario
Registro	idRegistro	Int	N	Y	Clave principal de la tabla
Registro	idTramite	Int	N	N	Trámite al que pertenece el registro
Registro	idFormulario	Int	Y	N	Id del Formulario que se ejecuta
RegistrosExternos	idRegistrosExternos	Int	N	Y	Clave principal de la tabla
RegistrosExternos	idTramite	Int	N	N	Id del trámite donde se ejecuta la integración
RegistrosExternos	idParametro	Int	N	N	Id del parametro de entrada
RegistrosExternos	dato	VarCharMax	N	N	Dato correspondiente al parametro de entrada
Tramite	idTramite	Int	N	Y	Clave principal de la tabla
Tramite	idInstancia	Int	N	N	Instancia a la que pertenece el trámite
Tramite	estadoEjecucion	NChar (1)	N	N	Estado del trámite
Tramite	fechaInicio	DateTime	Y	N	Fecha de inicio del trámite
Tramite	fechaLimite	Date	Y	N	Fecha límite del trámite
Tramite	fechaFin	DateTime	Y	N	Fecha de fin del trámite
Tramite	idVersionElemento	Int	Y	N	Versión del elemento del trámite
Tramite	idUsuario	Int	Y	N	Nuevo usuario responsable del trámite, cuando se realiza la reasignación

Elaboración: Autores

En el Cuadro 218 se presentan los campos pertenecientes a las tablas que conforman al esquema "tarea".

**Cuadro 218.** Columnas correspondientes a las tablas del esquema "tarea".

Tabla	Atributo	Tipo de dato	Nulo	PK	Descripción
CabeceraTarea	idCabeceraTarea	Int	N	Y	Clave principal de la tabla
CabeceraTarea	titulo	VarChar (100)	N	N	Título que se le da al proceso
CabeceraTarea	objetivoProceso	VarChar (500)	N	N	Objetivo del proceso
CabeceraTarea	fechaInicio	DateTime	N	N	Fecha de creación de la cabecera
CabeceraTarea	fechaFin	DateTime	Y	N	Fecha en la que finalizan todas las tareas
CabeceraTarea	fechaLimite	Date	N	N	Fecha límite para la entrega de las tareas
CabeceraTarea	idUsuarioAdministrador	Int	N	N	Usuario que crea las tareas
CabeceraTarea	listo	Bit	N	N	Establece que se ha realizado la asignación de recursos
CabeceraTarea	idUsuarioSupervisor	Int	N	N	Usuario que supervisa el desarrollo del proceso
CabeceraTarea	progreso	Numeric	N	N	Muestra el progreso de culminación de las tareas
CabeceraTarea	eliminado	Bit	N	N	Establece si una cabecera esta eliminada
CabeceraTarea	codigoUnico	VarCharMax	N	N	Código único de la cabecera (proceso)
CabeceraTarea	instanciado	Bit	N	N	Establece si el proceso ha sido instanciado
CabeceraTarea	rediseño	Bit	Y	N	Establece si el proceso está en etapa de rediseño
CampoTarea	idCampoTarea	Int	N	Y	Clave principal de la tabla
CampoTarea	campo	VarChar (1000)	N	N	Nombre del campo de la tarea
CampoTarea	idTipoDato	Int	N	N	Tipo de dato del campo
ControlCabecera	idControl	Int	N	N	Clave principal de la tabla
ControlCabecera	accion	VarChar	N	N	Especifica el tipo de

		(50)			sentencia sql se realizó
ControlCabecera	observacion	VarCharMax	N	N	Indica el motivo del registro
ControlCabecera	fecha	DateTime	N	N	Fecha en la que se crea el registro
ControlCabecera	idCabecera	Int	N	N	Id de la cabecera a la que pertenece el registro
ControllIntegracion	idControl	Int	N	Y	Clave principal de la tabla
ControllIntegracion	idIntegracion	Int	N	N	Id de la integración ejecutada
ControllIntegracion	codigo	VarChar (50)	N	N	Código del error originado
ControllIntegracion	detalle	VarCharMax	N	N	Detalle del error
ControllIntegracion	tipoIntegracion	NVarChar (2)	N	N	Tipo de integración ejecutada
DetalleTarea	idDetalleTarea	Int	N	Y	Clave principal de la tabla
DetalleTarea	registro	VarCharMax	N	N	Ruta donde se guarda la tarea
DetalleTarea	idCampoTarea	Int	N	N	Tipo de campo de la tarea
DetalleTarea	idTarea	Int	N	N	Tarea a la que pertenece el detalle
Formulario	idFormulario	Int	N	Y	Clave principal de la tabla
Formulario	idTarea	Int	N	N	Id de la tarea a la que pertenece el Formulario
Formulario	descripcion	VarChar (500)	N	N	Descripción del formulario
FormularioElemento	idFormularioElemento	Int	N	Y	Clave principal de la tabla
FormularioElemento	idFormulario	Int	N	N	Id del Formulario al que pertenece el elemento
FormularioElemento	idElemento	VarChar (150)	N	N	Id del Elemento
FormularioElemento	nombreElemento	VarChar (50)	N	N	Nombre del elemento
FormularioElemento	tipoElemento	VarChar (50)	N	N	Tipo del elemento
FormularioElemento	subTipo	VarChar (50)	Y	N	Subtipo del elemento
FormularioElemento	idIntegracion	Int	Y	N	Id de la integración asignada al elemento
FormularioElemento	integracionCampo	VarChar (500)	Y	N	Campo de la integración a mostrar en el elemento
FormularioElemento	clase	VarChar (50)	Y	N	Clase HTML del elemento
FormularioElemento	mascara	VarChar (50)	Y	N	Máscara asignada al elemento
FormularioElemento	tipoIntegracion	VarChar (50)	Y	N	Indica el tipo de integración asignada al elemento

<b>FormularioElemento</b>	value	VarCharMax	Y	N	Dato que se establece en el value de un elemento tipo select
<b>HistorialTarea</b>	idHistorial	Int	N	N	Clave principal de la tabla
<b>HistorialTarea</b>	fecha	DateTime	N	N	Fecha de creación del registro
<b>HistorialTarea</b>	idResponsableAnterior	Int	N	N	Id del Usuario anterior de la tarea
<b>HistorialTarea</b>	idResponsableActual	Int	N	N	Id del Usuario actual de la tarea
<b>HistorialTarea</b>	observacion	VarCharMax	N	N	Establece el motivo del cambio
<b>HistorialTarea</b>	idTarea	Int	N	N	Id de la tarea en la que se realiza el cambio
<b>HistorialTarea</b>	idCabecera	Int	N	N	Id de la cabecera a la que pertenece la tarea
<b>IntegracionExterna</b>	idIntegracionExterna	Int	N	Y	Clave principal de la tabla
<b>IntegracionExterna</b>	url	VarChar (1000)	N	N	Dirección URL del servicio web
<b>IntegracionExterna</b>	usuario	VarChar (200)	Y	N	Usuario del servicio
<b>IntegracionExterna</b>	contraseña	VarChar (200)	Y	N	Contraseña del servicio
<b>IntegracionExterna</b>	idTarea	Int	N	N	Id de la tarea correspondiente
<b>IntegracionInterna</b>	idIntegracionInterna	Int	N	Y	Clave principal de la tabla
<b>IntegracionInterna</b>	tipo	VarChar (200)	N	N	Establece si es una consulta simple o con filtros
<b>IntegracionInterna</b>	idFormulario	Int	N	N	Id del formulario del cual se obtendran los datos
<b>IntegracionInterna</b>	idTarea	Int	N	N	Id de la tarea correspondiente
<b>IntegracionInterna</b>	desde	Date	Y	N	Indica la fecha desde la cual se obtendran los datos
<b>IntegracionInterna</b>	hasta	Date	Y	N	Indica la fecha hasta la cual se obtendran los datos
<b>IntegracionInterna</b>	hastaUltimo	Bit	Y	N	Indica que se obtendra hasta el ultimo ingreso registrado
<b>IntegracionInterna</b>	desdelnicio	Bit	Y	N	Indica que se obtendra desde el primer ingreso registrado
<b>MensajeTarea</b>	idMensaje	Int	N	Y	Clave principal de la tabla
<b>MensajeTarea</b>	idTarea	Int	Y	N	Id de la tarea a a que corresponde el mensaje
<b>MensajeTarea</b>	idCabeceraTarea	Int	Y	N	Id de la cabecera a la

					que corresponde la tarea
MensajeTarea	mensaje	VarCharMax	N	N	Mensaje que se envía al usuario
MensajeTarea	alerta	Bit	N	N	Booleano que indica si el mensaje es una alerta
MensajeTarea	rutaTarea	VarCharMax	Y	N	Ruta de la tarea que se debe realizar
MensajeTarea	fecha	DateTime	N	N	Fecha en la que se envía el mensaje
MensajeTarea	estado	NVarChar (2)	N	N	Estado del mensaje
MensajeTarea	idUsuario	Int	N	N	Usuario al que se le enviará el mensaje
MensajeTarea	rol	NVarChar (256)	N	N	Rol del Usuario
MensajeTarea	idHistorial	Int	Y	N	Id del historial
MensajeTarea	automatico	Bit	Y	N	Establece si el mensaje es automático
ParametroIE	idParametro	Int	N	Y	Clave principal de la tabla
ParametroIE	idIntegracionExterna	Int	N	N	Id de la Integración a la que pertenece el parámetro
ParametroIE	campo	VarChar (500)	N	N	Nombre del campo
ParametroIE	tipoParametro	NVarChar (2)	N	N	Indica si el parámetro es de entrada o salida
ParametroIE	idTipoDato	Int	N	N	Id del tipo de dato
ParametroII	idParametro	Int	N	Y	Clave principal de la tabla
ParametroII	idIntegracionInterna	Int	N	N	Id de la Integración a la que pertenece el parámetro
ParametroII	idFormularioElemento	Int	N	N	Id del Formulario Elemento que se usa en la Integración
ParametroII	tipoParametro	NVarChar (2)	N	N	Indica si el parámetro es de entrada o salida
ReferenciaFormulario	idReferencia	Int	N	Y	Clave principal de la tabla
ReferenciaFormulario	idTarea	Int	N	N	Id de la tarea correspondiente
ReferenciaFormulario	idFormulario	Int	N	N	Id del formulario referenciado
ReferenciaFormulario	ultimoIngreso	Bit	Y	N	Indica si se obtiene el ultimo ingreso
ReferenciaFormulario	ingresoEspecifico	Bit	Y	N	Indica si se obtiene datos en una fecha especifica
ReferenciaFormulario	desdelnicio	Bit	Y	N	Indica que se obtienen los datos desde el inicio
ReferenciaFormulario	desde	Date	Y	N	Indica la fecha de inicio desde la cual se obtendran los datos

ReferenciaFormulario	hasta	Date	Y	N	Indica la fecha de fin hasta la cual se obtendrán los datos
ReferenciaFormulario	hastaUltimo	Bit	Y	N	Indica que se obtiene datos hasta el último ingreso
ReferenciaFormulario	especificoPeriodo	NVarChar (5)	Y	N	Establece un periodo de tiempo para obtener datos
ReferenciaFormularioElemento	idReferenciaFormElem	Int	N	Y	Clave principal de la tabla
ReferenciaFormularioElemento	idReferencia	Int	N	N	Id de la Referencia
ReferenciaFormularioElemento	idFormularioElemento	Int	N	N	Id del Formulario Elemento
ReferenciaFormularioElemento	editable	Bit	Y	N	Indica si el elemento es editable
ReferencialIntegracion	idReferencialIntegracion	Int	N	Y	Clave principal de la tabla
ReferencialIntegracion	idTarea	Int	N	N	Id de la tarea correspondiente
ReferencialIntegracion	rutaFormulario	VarCharMax	Y	N	Ruta del formulario
ReferencialIntegracion	idIntegracion	Int	N	N	Id de la integración utilizada
ReferencialIntegracion	tipoIntegracion	VarChar (50)	N	N	Tipo de integración utilizada
Tarea	idTarea	Int	N	Y	Clave principal de la tabla
Tarea	idCabeceraTarea	Int	N	N	Cabecera a la que pertenece la tarea
Tarea	titulo	VarChar (100)	N	N	Tema de la tarea
Tarea	descripcion	VarCharMax	N	N	Breve descripción de la tarea
Tarea	idUsuarioResponsable	Int	N	N	Usuario responsable de realizar la tarea
Tarea	estado	NVarChar (2)	N	N	Estado de la tarea
Tarea	fechaEnvio	DateTime	N	N	Fecha en que la tarea es enviada al usuario
Tarea	fechaEntrega	DateTime	Y	N	Fecha en la que el usuario entrega la tarea
Tarea	tipoTarea	NVarChar (5)	N	N	Establece el tipo de tarea
Tarea	fechaLimite	Date	N	N	Fecha límite para entregar la tarea
Tarea	eliminado	Bit	N	N	Indica si la tarea esta eliminada

Elaboración: Autores

En el Cuadro 219 se presentan los campos pertenecientes a las tablas que conforman al esquema “usuario”.

**Cuadro 219.** Columnas correspondientes a las tablas del esquema "usuario".

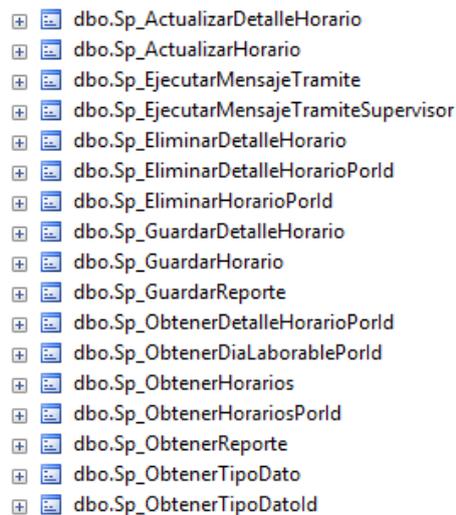
Tabla	Atributo	Tipo de dato	Nulo	PK	Descripción
Acceso	idAcceso	Int	N	Y	Clave principal de la tabla
Acceso	idUsuario	Int	N	N	Usuario que accede al sistema
Acceso	fecha	DateTime	N	N	Fecha y hora de ingreso al sistema
Acceso	tipo	VarChar (2)	N	N	Fecha y hora de salida del sistema
Acceso	ip	VarCharMax	N	N	Dirección IP de la máquina que accede al sistema
Acceso	mac	VarCharMax	N	N	Dirección MAC de la máquina que accede al sistema
Caracteristica	idCaracteristica	Int	N	Y	Clave principal de la tabla
Caracteristica	campo	VarChar (100)	N	N	Nombre de la característica
Caracteristica	idTipoDato	Int	N	N	Tipo de dato de la característica
Cargo	idCargo	Int	N	Y	Clave principal de la tabla
Cargo	nombre	VarChar (50)	N	N	Nombre del cargo
Cargo	eliminado	Bit	N	N	Indica si el cargo ha sido eliminado
Usuario	idUsuario	Int	N	Y	Clave principal de la tabla
Usuario	nombreCompleto	VarChar (100)	N	N	Nombre del usuario
Usuario	cedula	VarChar (13)	N	N	Cedula del usuario
Usuario	idIdentity	NVarChar (128)	N	N	Identity del usuario
Usuario	idCargo	Int	N	N	Id del cargo correspondiente
Usuario	eliminado	Bit	N	N	Establece si el usuario está eliminado
Usuario	idHorario	Int	N	N	Id del horario correspondiente al usuario
Usuario	restringido	Bit	Y	N	Establece si el usuario tiene acceso restringido
UsuarioCaracteristica	idUsuarioCaracteristica	Int	N	Y	Clave principal de la tabla
UsuarioCaracteristica	idUsuario	Int	N	N	Id del usuario
UsuarioCaracteristica	idCaracteristica	Int	N	N	Id de la característica
UsuarioCaracteristica	registro	VarChar (50)	N	N	Dato de la característica

Elaboración: Autores

## 1.19 PROCEDIMIENTOS ALMACENADOS

A continuación, se presentan los procedimientos almacenados (SP) establecidos según los esquemas de la BD, los cuales permitieron llevar a cabo las acciones del CRUD (Create (crear), Read (consultar), Update (actualizar) y Delete (eliminar)) para las entidades y otras operaciones con métodos propios de SQL, tales como: obtener fecha actual (GetDate).

En la Figura 53 se presentan los SP que utilizan las tablas o entidades que no pertenecen a ningún esquema personalizado, por lo tanto tienen el sufijo (.dbo) por defecto del SQL. Por otra parte, en la Figura 54 se muestran los SP que realizan diferentes acciones para las entidades pertenecientes al esquema “diseñoproceso”, en la Figura 55 los SP referentes a “instanciaproceso”, en la Figura 56 de “tarea” y Figura 57 de “usuario”.



- + [document icon] dbo.Sp\_ActualizarDetalleHorario
- + [document icon] dbo.Sp\_ActualizarHorario
- + [document icon] dbo.Sp\_EjecutarMensajeTramite
- + [document icon] dbo.Sp\_EjecutarMensajeTramiteSupervisor
- + [document icon] dbo.Sp\_EliminarDetalleHorario
- + [document icon] dbo.Sp\_EliminarDetalleHorarioPorId
- + [document icon] dbo.Sp\_EliminarHorarioPorId
- + [document icon] dbo.Sp\_GuardarDetalleHorario
- + [document icon] dbo.Sp\_GuardarHorario
- + [document icon] dbo.Sp\_GuardarReporte
- + [document icon] dbo.Sp\_ObtenerDetalleHorarioPorId
- + [document icon] dbo.Sp\_ObtenerDiaLaborablePorId
- + [document icon] dbo.Sp\_ObtenerHorarios
- + [document icon] dbo.Sp\_ObtenerHorariosPorId
- + [document icon] dbo.Sp\_ObtenerReporte
- + [document icon] dbo.Sp\_ObtenerTipoDato
- + [document icon] dbo.Sp\_ObtenerTipoDatold

**Figura 53.** Procedimientos almacenados que no pertenecen a ningún esquema.

**Elaboración:** Autores

+	🔍	diseñoproceso.Sp_ContarProcesos	+	🔍	diseñoproceso.Sp_ObtenerFlujoAtributoXAtributo
+	🔍	diseñoproceso.Sp_ContarVersionesxProceso	+	🔍	diseñoproceso.Sp_ObtenerFlujoAtributoXId
+	🔍	diseñoproceso.Sp_DefinirTipoCompuerta	+	🔍	diseñoproceso.Sp_ObtenerIdProceso
+	🔍	diseñoproceso.Sp_EjecutarGuardarVersion	+	🔍	diseñoproceso.Sp_ObtenerModeloXCabecera
+	🔍	diseñoproceso.Sp_EliminarAtributoElemento	+	🔍	diseñoproceso.Sp_ObtenerModeloXCodigoUnico
+	🔍	diseñoproceso.Sp_EliminarCoordenada	+	🔍	diseñoproceso.Sp_ObtenerObjetoFlujoPorId
+	🔍	diseñoproceso.Sp_GuardarAtributos	+	🔍	diseñoproceso.Sp_ObtenerProcesoPorId
+	🔍	diseñoproceso.Sp_GuardarCoordenadas	+	🔍	diseñoproceso.Sp_ObtenerProcesos
+	🔍	diseñoproceso.Sp_GuardarElementos	+	🔍	diseñoproceso.Sp_ObtenerProcesosXCabecera
+	🔍	diseñoproceso.Sp_GuardarProceso	+	🔍	diseñoproceso.Sp_ObtenerReglad
+	🔍	diseñoproceso.Sp_GuardarRegla	+	🔍	diseñoproceso.Sp_ObtenerUltimoProceso
+	🔍	diseñoproceso.Sp_GuardarVersion	+	🔍	diseñoproceso.Sp_ObtenerVersionElementoPorId
+	🔍	diseñoproceso.Sp_GuardarVersionElemento	+	🔍	diseñoproceso.Sp_ObtenerVersionesCodigoProceso
+	🔍	diseñoproceso.Sp_ModificarAtributoElemento	+	🔍	diseñoproceso.Sp_ObtenerVersionesProceso
+	🔍	diseñoproceso.Sp_ModificarCoordenadaRegla	+	🔍	diseñoproceso.Sp_ObtenerVersionPorElemento
+	🔍	diseñoproceso.Sp_ModificarVersionElemento	+	🔍	diseñoproceso.Sp_ObtenerVersionPorId
+	🔍	diseñoproceso.Sp_ObtenerAtributos	+	🔍	diseñoproceso.Sp_ProcesoListo
+	🔍	diseñoproceso.Sp_ObtenerAtributosPorCadaElemento	+	🔍	diseñoproceso.Sp_RediseñoProceso
+	🔍	diseñoproceso.Sp_ObtenerAtributosPorVersion	+	🔍	diseñoproceso.Sp_VerificarInstanciasProceso
+	🔍	diseñoproceso.Sp_ObtenerCodigoProcesoVersion	+	🔍	diseñoproceso.Sp_VerificarProceso
+	🔍	diseñoproceso.Sp_ObtenerCodigosConFinMensaje			
+	🔍	diseñoproceso.Sp_ObtenerCodigosConInicioMensaje			
+	🔍	diseñoproceso.Sp_ObtenerCoordenadaPorId			
+	🔍	diseñoproceso.Sp_ObtenerCoordenadaPorOrigen			
+	🔍	diseñoproceso.Sp_ObtenerCoordenadaPorVersionElemento			
+	🔍	diseñoproceso.Sp_ObtenerElementoPorCodigo			
+	🔍	diseñoproceso.Sp_ObtenerElementoPorId			
+	🔍	diseñoproceso.Sp_ObtenerElementoPorProceso			
+	🔍	diseñoproceso.Sp_ObtenerElementoPorVersion			

**Figura 54.** Procedimientos almacenados pertenecientes al esquema "diseñoproceso".

**Elaboración:** Autores

+ [icon]	instanciaproceso.Sp_CambiarEstadoMensaje	+ [icon]	instanciaproceso.Sp_ObtenerCabeceraPorProceso
+ [icon]	instanciaproceso.Sp_CambiarEstadoMensajeSupervisor	+ [icon]	instanciaproceso.Sp_ObtenerCoordenadaPorIdCoordenada
+ [icon]	instanciaproceso.Sp_ContinuarInstancia	+ [icon]	instanciaproceso.Sp_ObtenerDatoPorIdTramite
+ [icon]	instanciaproceso.Sp_EjecucionTramiteTimer	+ [icon]	instanciaproceso.Sp_ObtenerDetallesRegistro
+ [icon]	instanciaproceso.Sp_EjecutarErrorInstancia	+ [icon]	instanciaproceso.Sp_ObtenerHistorialPorInstancia
+ [icon]	instanciaproceso.Sp_EjecutarFinTerminal	+ [icon]	instanciaproceso.Sp_ObtenerIdElemFormPorIdReferenciaElemForm
+ [icon]	instanciaproceso.Sp_EjecutarProcesoTimer	+ [icon]	instanciaproceso.Sp_ObtenerIdInstanciaPorTramite
+ [icon]	instanciaproceso.Sp_EjecutarTramite	+ [icon]	instanciaproceso.Sp_ObtenerIdProcesoPorVersion
+ [icon]	instanciaproceso.Sp_EliminarAdjuntoPorIdInstancia	+ [icon]	instanciaproceso.Sp_ObtenerIdTareaPorIdTramite
+ [icon]	instanciaproceso.Sp_EliminarDetalleRegistro	+ [icon]	instanciaproceso.Sp_ObtenerIdVersionPorElemento
+ [icon]	instanciaproceso.Sp_EliminarInstancia	+ [icon]	instanciaproceso.Sp_ObtenerIdVersionPorInstancia
+ [icon]	instanciaproceso.Sp_EliminarRegistro	+ [icon]	instanciaproceso.Sp_ObtenerInstanciaPorId
+ [icon]	instanciaproceso.Sp_EliminarRegistrosExternos	+ [icon]	instanciaproceso.Sp_ObtenerInstancias
+ [icon]	instanciaproceso.Sp_EstablecerInstanciaEspera	+ [icon]	instanciaproceso.Sp_ObtenerInstanciasPorCodUnico
+ [icon]	instanciaproceso.Sp_FinalizarEstadoMensaje	+ [icon]	instanciaproceso.Sp_ObtenerInstanciasPorProceso
+ [icon]	instanciaproceso.Sp_FinalizarInstancia	+ [icon]	instanciaproceso.Sp_ObtenerInstanciasPorUsuarioR
+ [icon]	instanciaproceso.Sp_GuardarAdjunto	+ [icon]	instanciaproceso.Sp_ObtenerMensajeReinicio
+ [icon]	instanciaproceso.Sp_GuardarControlIntegracion	+ [icon]	instanciaproceso.Sp_ObtenerMensajeTramite
+ [icon]	instanciaproceso.Sp_GuardarDetalleRegistro	+ [icon]	instanciaproceso.Sp_ObtenerMensajeTramiteIdentity
+ [icon]	instanciaproceso.Sp_GuardarEjecucionSubproceso	+ [icon]	instanciaproceso.Sp_ObtenerMensajeTramiteIdentityRol
+ [icon]	instanciaproceso.Sp_GuardarHistorialInstancia	+ [icon]	instanciaproceso.Sp_ObtenerMensajeTramitePorInstancia
+ [icon]	instanciaproceso.Sp_GuardarInstancia	+ [icon]	instanciaproceso.Sp_ObtenerPorIdTramite
+ [icon]	instanciaproceso.Sp_GuardarMensajeTramite	+ [icon]	instanciaproceso.Sp_ObtenerProcesoSimplePorUsuario
+ [icon]	instanciaproceso.Sp_GuardarMensajeTramiteSupervisor	+ [icon]	instanciaproceso.Sp_ObtenerRegistroExternoldParametro
+ [icon]	instanciaproceso.Sp_GuardarRegistro	+ [icon]	instanciaproceso.Sp_ObtenerRegistrosPorIdTramite
+ [icon]	instanciaproceso.Sp_GuardarRegistroExterno	+ [icon]	instanciaproceso.Sp_ObtenerRutaAdjuntoPorIdFormularioElemento
+ [icon]	instanciaproceso.Sp_GuardarTramite	+ [icon]	instanciaproceso.Sp_ObtenerSubProcesoPorInstancia
+ [icon]	instanciaproceso.Sp_ObtenerAdjuntoPorInstancia	+ [icon]	instanciaproceso.Sp_ObtenerSupervisorPorInstancia
+ [icon]	instanciaproceso.Sp_ObtenerTramitePorId	+ [icon]	instanciaproceso.Sp_ObtenerTramiteAnterior
+ [icon]	instanciaproceso.Sp_ObtenerTramitePorIdentity		
+ [icon]	instanciaproceso.Sp_ObtenerTramiteSiguiente		
+ [icon]	instanciaproceso.Sp_ObtenerTramitesPorCoordenada		
+ [icon]	instanciaproceso.Sp_ObtenerTramitesPorElemento		
+ [icon]	instanciaproceso.Sp_ObtenerTramitesPorIdInstancia		
+ [icon]	instanciaproceso.Sp_ObtenerTramitesPorUsuario		
+ [icon]	instanciaproceso.Sp_ObtenerTramitesPorUsuarioR		
+ [icon]	instanciaproceso.Sp_ObtenerUsuarioInicInstancia		
+ [icon]	instanciaproceso.Sp_ObtenerUsuarioSupervisorPorProceso		
+ [icon]	instanciaproceso.Sp_ReasignarResponsable		
+ [icon]	instanciaproceso.Sp_ReiniciarInstancia		
+ [icon]	instanciaproceso.Sp_ReiniciarTramite		
+ [icon]	instanciaproceso.Sp_SupervisiónDeTramites		
+ [icon]	instanciaproceso.Sp_VerificarInstanciaSubproceso		
+ [icon]	instanciaproceso.Sp_VerificarSubProcesosDeInstancia		
+ [icon]	instanciaproceso.SP_VerificarYEjecutarTamiteSiguiente		

**Figura 55.** Procedimientos almacenados pertenecientes al esquema "instanciaproceso".

**Elaboración:** Autores

+  tarea.Sp_AceptarORechazarTarea	+  tarea.Sp_ObtenerCabeceraTareasId
+  tarea.Sp_ActualizarFechaTareaRediseño	+  tarea.Sp_ObtenerCabeceraTareasIdentityYRol
+  tarea.Sp_ActualizarFechaYResponsable	+  tarea.Sp_ObtenerCabeceraTareaUnico
+  tarea.Sp_EliminarCabecera	+  tarea.Sp_ObtenerCampoTarea
+  tarea.Sp_EliminarReferenciaElemento	+  tarea.Sp_ObtenerCampoTareald
+  tarea.Sp_EliminarTarea	+  tarea.Sp_ObtenerCincoAlertas
+  tarea.Sp_EstablecerFechaLimiteProcesoRediseño	+  tarea.Sp_ObtenerCincoAlertasTramite
+  tarea.Sp_GuardarCabecera	+  tarea.Sp_ObtenerControlCabecera
+  tarea.Sp_GuardarCampoTarea	+  tarea.Sp_ObtenerDetallesDeElementos
+  tarea.Sp_GuardarControlCabecera	+  tarea.Sp_ObtenerDetalleTarea
+  tarea.Sp_GuardarFormulario	+  tarea.Sp_ObtenerDetalleTareaPorIdTarea
+  tarea.Sp_GuardarFormularioElemento	+  tarea.Sp_ObtenerFormularioElemento
+  tarea.Sp_GuardarIntegracionExterna	+  tarea.Sp_ObtenerFormularios
+  tarea.Sp_GuardarIntegracionInterna	+  tarea.Sp_ObtenerFormulariosPorCabecera
+  tarea.Sp_GuardarMensajeAdministrador	+  tarea.Sp_ObtenerFormulariosPorTarea
+  tarea.Sp_GuardarMensajeSupervisor	+  tarea.Sp_ObtenerHistorialPorCabecera
+  tarea.Sp_GuardarMensajeTarea	+  tarea.Sp_ObtenerIdTramiteYFechas
+  tarea.Sp_GuardarParametrosIE	+  tarea.Sp_ObtenerIEPorId
+  tarea.Sp_GuardarParametrosII	+  tarea.Sp_ObtenerIEPorTarea
+  tarea.Sp_GuardarReferenciaFormulario	+  tarea.Sp_ObtenerIntegraciones
+  tarea.Sp_GuardarReferenciaFormularioElemento	+  tarea.Sp_ObtenerIntegracionInternald
+  tarea.Sp_GuardarReferencialIntegracion	+  tarea.Sp_ObtenerMensajesAlertaPorIdentity
+  tarea.Sp_GuardarTarea	+  tarea.Sp_ObtenerMensajesPorIdentity
+  tarea.Sp_ModificarEstadoMensaje	+  tarea.Sp_ObtenerMensajesPorIdTarea
+  tarea.Sp_ModificarEstadoMensajeEnviado	+  tarea.Sp_ObtenerMensajesPorIdUsuario
+  tarea.Sp_ModificarEstadoMensajeSupervisor	+  tarea.Sp_ObtenerMensajeTarea
+  tarea.Sp_ModificarFechaLimiteProceso	+  tarea.Sp_ObtenerParametrosPorIE
+  tarea.Sp_ModificarMensajeTarea	+  tarea.Sp_ObtenerParametrosPorII
+  tarea.Sp_ObtenerReferenciaFormularioPorTarea	+  tarea.Sp_ObtenerReferenciaFormElementoPorForm
+  tarea.Sp_ObtenerReferencialIntegracionXTarea	
+  tarea.Sp_ObtenerRegistrosDeFormElemento	
+  tarea.Sp_ObtenerTareas	
+  tarea.Sp_ObtenerTareasIdentityRolYCabecera	
+  tarea.Sp_ObtenerTareasIdentityYRol	
+  tarea.Sp_ObtenerTareasNoEntregadas	
+  tarea.Sp_ObtenerTareasPorId	
+  tarea.Sp_ObtenerTareasPorIdCabecera	
+  tarea.Sp_ObtenerTareasPorIdUsuario	
+  tarea.Sp_RealizarTarea	

**Figura 56.** Procedimientos almacenados pertenecientes al esquema "tarea".

**Elaboración:** Autores

- + usuario.Sp\_ActualizarCargos
- + usuario.Sp\_ActualizarUsuario
- + usuario.Sp\_EliminarCaracteristica
- + usuario.Sp\_EliminarCargo
- + usuario.Sp\_EliminarUsuario
- + usuario.Sp\_EliminarUsuarioCaracteristica
- + usuario.Sp\_GuardarAcceso
- + usuario.Sp\_GuardarCarateristica
- + usuario.Sp\_GuardarCargos
- + usuario.Sp\_GuardarUsuarioCaracteristica
- + usuario.Sp\_GuardarUsuarios
- + usuario.Sp\_ModificarCaracteristica
- + usuario.Sp\_ModificarUsuarioCaracteristica
- + usuario.Sp\_ObtenerAccesoPorIdUsuario
- + usuario.Sp\_ObtenerCaracteristica
- + usuario.Sp\_ObtenerCaracteristicaId
- + usuario.Sp\_ObtenerCargos
- + usuario.Sp\_ObtenerCargosConUsuarios
- + usuario.Sp\_ObtenerCargosId
- + usuario.Sp\_ObtenerUsuario
- + usuario.Sp\_ObtenerUsuarioCaracteristicaPorUsuario
- + usuario.Sp\_ObtenerUsuarioPorCargo
- + usuario.Sp\_ObtenerUsuarioPorId
- + usuario.Sp\_ObtenerUsuarioPorIdentity
- + usuario.Sp\_ObtenerUsuarioPorRol
- + usuario.Sp\_ObtenerUsuariosOperadores

**Figura 57.** Procedimientos almacenados pertenecientes al esquema "usuario".  
**Elaboración:** Autores

## 1.20 ROLES Y USUARIOS

A fin de implementar seguridad para el acceso a la BD del sistema, fue necesario el uso de credenciales establecidas en el Cuadro 220.

**Cuadro 220.** Credenciales de acceso a la BD Motor\_Workflow.

Base de datos	Rol	Usuario	Contraseña
Motor_Workflow	Súper administrador	sa	12345

**Elaboración:** Autores

## GLOSARIO Y ABREVIATURAS

### 1.21 GLOSARIO

- **ADO.NET Entity Framework:** permite a los desarrolladores crear aplicaciones de acceso a datos programando con un modelo de

aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional.

- **ApiController:** retorna datos que serán serializados y enviados al cliente.
- **API RestFul:** forma de nombrar a aquellas APIs que si cumplen de forma correcta con la definición de REST.
- **ASP.NET:** marco web de código abierto para la creación de aplicaciones y servicios web modernos con .NET.
- **C#:** (pronunciado C Sharp) es un lenguaje de programación bajo el paradigma POO, desarrollado por Microsoft, derivado de C/C++ y Java.
- **Dirección IP:** número que identifica, de manera lógica y jerárquica, a una Interfaz en red (elemento de comunicación/conexión) de un dispositivo (computadora, tableta) que utilice el protocolo IP.
- **Dirección MAC:** identificador único asignado por el fabricante a una pieza de hardware de red.
- **Entity Data Model o simplemente Entity Model:** es una especificación que permite definir los datos que usan las aplicaciones integradas en Entity Framework.
- **Framework:** conjunto de componentes, herramientas y documentación que facilita el desarrollo de software.
- **Microsoft:** empresa estadounidense dedicada al sector de software y el hardware.
- **Núcleo:** software que se encarga del seguimiento de los casos o instancias de los procesos implementados en el Workflow.
- **REST:** técnica de arquitectura de software, es decir, un conjunto de principios y patrones de comunicación que ayudan a crear una forma de pensar y construir las APIs.
- **Workflow:** automatización de un proceso de negocio, en todo o en parte, durante el cual los Documentos, Información o las Tareas (Actividades) se pasan de un participante a otro para que realicen alguna acción, de acuerdo a un conjunto de reglas procedimentales.

## 1.22 ABREVIATURAS

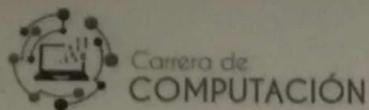
- **ANSI:** American National Standards Institute.
- **API:** Application Programming Interface.
- **BD:** Base de datos.
- **CRUD:** por sus siglas en inglés de los métodos crear (Create), leer (Read), actualizar (Update) y eliminar (Delete).
- **ESPAM MFL:** Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.
- **HTTP:** Hypertext Transfer Protocol.
- **IP:** Internet Protocol
- **MAC:** Media Access Control
- **MEIDAW:** Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow.
- **PK:** Primary Key.
- **RAM:** Random Access Memory.
- **REST:** Representational State Transfer.
- **SGBD:** Sistema Gestor de Base de Datos.
- **SP:** Stored Procedure.
- **SQL:** Structured Query Language.
- **UPS:** Unidad de Producción de Software.
- **VS:** Visual Studio.
- **XML:** Extensible Markup Language.

## BIBLIOGRAFÍA

- Arbeláez, O; Medina, F; Chaves, O. 2011. Herramientas para el desarrollo rápido de aplicaciones web. Pereira, CO. Scientia Et Technica. Vol. 17. p 254 - 258.
- Fotache, M. & Strimbei, C. 2015. SQL and Data Analysis. Some Implications for Data Analysis and Higher Education. RO. Journal Procedia Economics and Finance. Vol. 20. p 243-251.
- Martín, M; Leyva, G; Peña, N; Labrada, R. 2016. Proceso de réplica de datos con Microsoft SQL Server para el Replicador de Datos Reko. CU. Revista Cubana de Ciencias Informáticas. Vol. 10. p 128-142.

Microsoft. 2015. ASP.NET Web API. (En línea). Consultado, 12 de Oct. 2016. Formato HTML. Disponible en <https://msdn.microsoft.com>

Stein, M. & Geyer-Schulz, A. 2013. A Comparison of Five Programming Languages in a Graph Clustering Scenario. Karlsruhe, DE. Journal Universal Computer Science. Vol. 19. p 428-456.

**ANEXO 8 – CERTIFICACIÓN DEL TUTOR**Carrera de  
**COMPUTACIÓN**

Calceta, 18 de septiembre de 2017

Ingeniero  
**Daniel Agustín Mera Martínez, Mg.**  
Presidente del Tribunal de Hardware  
Carrera de Computación - ESPAM MFL

**Asunto:** CARTA AVAL

En calidad de Tutor de la Tesis titulada:

“NÚCLEO DE UN WORKFLOW DE PROCESOS ADMINISTRATIVOS EN LA ESPAM MFL”

De los postulantes **Dayana Helen Bailón Delgado y José Simón Mera Cantos** de la Carrera de Computación, considero que cumple con los requerimientos metodológicos y aportes científicos-técnicos suficientes, además de haber desarrollado los objetivos planteados. Motivo por el cual postulo y avalo la presente tesis para ser sometida a evaluación del Tribunal de Hardware, para su correspondiente revisión, sustentación y calificación.

Para los fines legales pertinentes, me suscribo a usted.

Atentamente,



Mgtr. Joffre Moreira Pico  
Carrera de Computación  
ESPAM MFL  
Correo: jmoreira@espam.edu.ec

*D. Mera*  
02/10/2017

CC: Ing. Luis Alberto Ortega Arcia  
**DIRECTOR DE LA CARRERA DE COMPUTACIÓN**

## ANEXO 8 – CERTIFICACIÓN DEL TRIBUNAL

**TRIBUNAL DE TESIS**  
LÍNEA DE INVESTIGACIÓN:  
**COMPUTACIÓN, ORGANIZACIONAL Y ALGORITMOS**

 Carrera de  
**COMPUTACIÓN**

Calceta, 10 de octubre del 2017

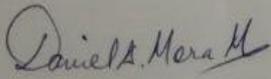
Señores  
Dayana Helen Bailón Delgado y José Simón Mera Cantos

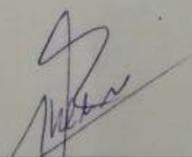
**POSTULANTES**

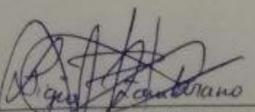
De nuestras consideraciones:

Los miembros del Tribunal, después de revisar las correcciones del TRABAJO DE TITULACIÓN Y ARTÍCULO CIENTÍFICO, PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN INFORMÁTICA cuyo tema es: **"NÚCLEO DE UN WORKFLOW DE PROCESOS ADMINISTRATIVOS EN LA ESPAM MFL"**, presentado por ustedes les informamos que luego de haber comprobado su cumplimiento y realizado los correctivos del caso, en la sesión correspondiente, se determinó que ha sido aprobado unanimidad en su contenido y forma, de acuerdo al reglamento vigente de Unidad de Titulación y a la viabilidad técnica y económica.

Atentamente:

  
Ing. Daniel A. Mera Martínez. Mg.  
**PRESIDENTE DEL TRIBUNAL**

  
Ing. Marlon R. Navia Mendoza. Mg.  
**MIEMBRO DEL TRIBUNAL**

  
Ing. Ligia E. Zambrano Solórzano. Mg.  
**SECRETARIA DEL TRIBUNAL**