



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

CARRERA INFORMÁTICA

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

TEMA:

**MÓDULO WEB PARA EL CONTROL DE ASISTENCIA DE LOS
ESTUDIANTES DE LA ESCUELA SUPERIOR POLITÉCNICA
AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ**

AUTORES:

**MARCOS CRISTHIAN ANZULES REYNA
LINCOLN EDUARDO SANTANDER ALCÍVAR**

TUTOR:

ING. ANGEL ALBERTO VÉLEZ MERO, MGS

CALCETA, NOVIEMBRE 2015

DERECHOS DE AUTORÍA

Anzules Reyna Marcos Cristhian y Santander Alcívar Lincoln Eduardo, 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.

.....

MARCOS C. ANZULES REYNA

.....

LINCOLN E. SANTANDER ALCÍVAR

CERTIFICACIÓN DEL TUTOR

Ángel Alberto Vélez Mero certifica haber tutelado la tesis **MÓDULO WEB PARA EL CONTROL DE ASISTENCIA DE LOS ESTUDIANTES DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ**, que ha sido desarrollada por Marcos Cristhian Anzules Reyna y Lincoln Eduardo Santander Alcívar, 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. ÁNGEL A.VÉLEZ MERO

APROBACIÓN DEL TRIBUNAL

Los suscritos integrantes del tribunal correspondiente, declaran que han **APROBADO** la tesis **MÓDULO WEB PARA EL CONTROL DE ASISTENCIA DE LOS ESTUDIANTES DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ**, que ha sido propuesta, desarrollada y sustentada por Marcos Cristhian Anzules Reyna y Lincoln Eduardo Santander Alcívar, 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
MIEMBRO

.....
ING. ORLANDO AYALA PULLAS
MIEMBRO

.....
ING. DANIEL A. MERA MARTÍNEZ
PRESIDENTE

AGRADECIMIENTO

Nuestros agradecimientos a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López por habernos brindado la oportunidad de poder formarnos como profesionales dentro de sus aulas y a través de las cuales logró impartir una educación de excelente calidad forjando el conocimiento y el carácter de sus estudiantes diariamente.

A Dios por darnos la vida y permitiéndonos estar aquí, felices y fuertes en todo sentido y por estar acompañándonos durante la realización de este trabajo.

A nuestros padres quienes gracias a sus esfuerzos y sacrificios hemos podido estudiar y encontrarnos a punto de culminar estudios como profesionales de tercer nivel, sin duda ellos mediante sus enseñanzas, nos han inculcado que el que persevera alcanza y el logro de hoy, y de los que vendrán, serán gracias a ellos.

A los profesores por habernos compartido sus conocimientos y buenas prácticas profesionales durante todo el ciclo de estudio y por su apoyo y comprensión en los momentos necesarios.

Y a todas las personas que estuvieron indirectamente relacionadas y contribuyeron de alguna manera para que la realización de este trabajo se llevara a cabo.

Los Autores

DEDICATORIA

Dedico este trabajo de tesis en primer lugar a Dios por permitirme estar con vida, por darme la fortaleza para no desistir y poder cumplir con este sueño.

A mis padres por haber sido un pilar fundamental en mi vida y apoyarme en todo momento de manera incondicional durante esta etapa.

A mis mejores amigos que en todo momento estuvieron pendientes de cómo iba en el proceso de la realización de este trabajo.

Para ellos es esta dedicatoria de tesis, pues es a ellos a quienes se las debo por su apoyo incondicional

Y a todos aquellos que no creyeron en mí, a aquellos que esperaban mi fracaso en cada paso que daba hacia la culminación de mis estudios, a aquellos que nunca esperaban que lograra terminar la carrera, a todos aquellos que apostaban a que me rendiría a medio camino, a todos los que supusieron que no lo lograría, a todos ellos también les dedico este trabajo.

.....

Lincoln E. Santander Alcívar

DEDICATORIA

Ofrezco este trabajo a las personas de las que siempre tendré un apoyo incondicional, que han reconocido mi esfuerzo, me han dado palabras de aliento en todo momento, en aquellos momentos en que el péndulo de la vida se dirige al lado de las desgracias, me hicieron desfallecer y perder la senda del porvenir, pero que con una palabra de aliento, pequeños detalles o un simple “yo creo en ti”, hicieron que retomara el rumbo del éxito.

Una de estas grandes personas es mi Madre, que con sus preocupaciones, siempre me daba un empujón para seguir alcanzando mis objetivos.

.....

Marcos C. Anzules Reyna

CONTENIDO GENERAL

CARÁTULA	i
DERECHOS DE AUTORÍA	ii
CERTIFICACIÓN DEL TUTOR	iii
APROBACIÓN DEL TRIBUNAL.....	iv
AGRADECIMIENTO.....	v
DEDICATORIA.....	vi
DEDICATORIA.....	vii
CONTENIDO GENERAL.....	viii
CONTENIDO DE CUADROS Y FIGURAS.....	xi
RESUMEN	xvii
PALABRAS CLAVES	xvii
ABSTRACT	xviii
KEY WORDS	xviii
CAPÍTULO I. ANTECEDENTES	19
1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA.....	19
1.2. JUSTIFICACIÓN	21
1.3. OBJETIVOS	23
1.3.1 OBJETIVO GENERAL.....	23
1.3.2 OBJETIVOS ESPECÍFICOS	23
1.4. HIPÓTESIS, PREMISA Y/O IDEAS A DEFENDER	24
1.4.1. IDEAS A DEFENDER.....	24
CAPÍTULO II. MARCO TEÓRICO.....	25
2.1. SISTEMA DE EDUCACIÓN SUPERIOR EDUCATIVO.....	25
2.1.1. UNIVERSIDADES Y ESCUELAS POLITÉCNICAS	25
2.2. PROCESO	26
2.3. OPTIMIZACIÓN	27
2.3.1. OPTIMIZACIÓN DE APLICACIONES	27
2.4. CONTROL DE GESTIÓN.....	27
2.4.1. CONTROL DE ASISTENCIA.....	28
2.5. PORTAFOLIO EDUCATIVO	29
2.5.1. PORTAFOLIO DEL PROFESOR	30
2.5.2. PORTAFOLIO ELECTRÓNICO	30
2.6. INTERNET	31
2.6.1. CONCEPTOS BÁSICOS DE INTERNET	31
2.6.2. FUNDAMENTOS DE LA WEB	31

2.6.3. WEB 2.0	32
2.6.4. NAVEGADOR WEB	32
2.7. SERVIDORES WEB.....	33
2.7.1. DIRECCIONES IP O DIRECCIONES INTERNET	33
2.7.2. DOMINIOS	34
2.7.3. PROVEEDORES DE SERVICIOS DE INTERNET, ISP (INTERNET SERVICE PROVIDER).....	34
2.7.4. SERVIDORES DNS (DOMAIN NAME SERVER - SERVIDOR DE NOMBRES DE DOMINIO)	35
2.8. CLIENTE	36
2.9. APLICACIÓN WEB	36
2.10. MÓDULO WEB	37
2.11. SERVICIO	38
2.11.1. SERVICIOS WEB.....	38
2.12. BOOTSTRAP	40
2.12.1. CARACTERÍSTICAS PRINCIPALES DE BOOTSTRAP	40
2.13. REPORTE	41
2.14. PRESENTACIÓN	41
2.14.1. PRESENTACIÓN DE REPORTES	42
2.15. BASES DE DATOS	42
2.15.1. BASE DE DATOS.....	42
2.15.2. SISTEMA GESTOR DE BASES DE DATOS	43
2.16. LENGUAJES DE PROGRAMACIÓN PARA LA WEB	44
2.16.1. ASP	44
2.16.2. ASP.NET	45
2.16.3. INTRODUCCIÓN A VISUAL STUDIO	46
2.17. METODOLOGÍA DE DESARROLLO MIDAS	46
2.17.1. MODELO INDEPENDIENTE DE LA COMPUTACIÓN (CIM)	49
2.17.2. MODELO INDEPENDIENTE DE PLATAFORMA.....	50
2.17.3. MODELO ESPECÍFICO DE LA PLATAFORMA (PSM).....	50
2.18. INVESTIGACIÓN DOCUMENTAL	51
2.19. ESTANDAR IEEE 830.....	51
CAPÍTULO III. DESARROLLO METODOLÓGICO	52
3.1. MÉTODO DE INVESTIGACIÓN DOCUMENTAL.....	52
3.2. METODOLOGÍA MIDAS	53
3.2.1. PRIMERA ITERACIÓN (MODELOS INDEPENDIENTES DE COMPUTACIÓN)	53

3.2.2. SEGUNDA ITERACIÓN (MODELO INDEPENDIENTES DE LA PLATAFORMA).....	54
3.2.3. MODELO ESPECÍFICO DE LA PLATAFORMA.....	55
3.3. DESARROLLO DEL SISTEMA	69
3.3.1. PARADIGMA DE DESARROLLO POR CAPAS.....	69
3.4. VERIFICACION DEL SISTEMA	70
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	71
4.1. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (ERS):.....	71
4.1.1. PROPÓSITO	71
4.1.2. DESCRIPCIÓN GENERAL	74
4.1.3. REQUERIMIENTOS ESPECÍFICOS.....	80
4.2. METODOLOGIA MIDAS	92
4.2.1. MODELO INDEPENDIENTE DE COMPUTACIÓN	92
4.2.2. MODELO INDEPENDIENTES DE LA PLATAFORMA.....	95
4.2.3. MODELO ESPECÍFICO DE LA PLATAFORMA.....	140
4.3. DESARROLLO DEL SISTEMA	144
4.3.1. CAPA DE DATOS	144
4.3.2. CAPA DE NEGOCIO.....	145
4.3.3. CAPA DE PRESENTACIÓN	146
4.4. VERIFICACIÓN DEL SISTEMA	146
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	150
5.1. CONCLUSIONES.....	150
5.2. RECOMENDACIONES	151
BIBLIOGRAFÍA	152
ANEXOS	157
ANEXO 1.....	158
ANEXO 2.....	161
ANEXO 3.....	168
ANEXO 4.....	189

CONTENIDO DE CUADROS Y FIGURAS

FIGURAS

Figura 2.1. Acciones que implica el portafolio electrónico	30
Figura 2.2. Red mundial de ordenadores	31
Figura 2.3. Estructura de una dirección web	33
Figura 2.4. Búsqueda de la IP en el servidor DNS	35
Figura 2.5. Devolución de la web solicitada	36
Figura 2.6. Esquema básico de una aplicación web.....	37
Figura 2.7. Arquitectura de MIDAS simplificada	47
Figura 2.8. Diagrama UML de proceso de un proyecto	49
Figura 2.9. Iteración del Modelo de Independiente de plataforma.....	50
Figura 3.10. Tipos de Datos de los Servicios Web del Módulo CASYS	58
Figura 3.11. Mensajes que intercambian las funciones.....	60
Figura 3.12. Configuración de las operaciones permitidas en los servicios SOAP.	62
Figura 3.13. Especificación de los protocolos utilizados.....	65
Figura 3.14. Servicios Web del Módulo de CASYS.	67
Figura 3.15. Visión de la Estructura de los Servicio Web.	68
Figura 4.16. Datos del banco Lógico.	91
Figura 4.17. Modelo de Dominio.....	94
Figura 4.18. Modelo de Negocio.....	95
Figura 4.19. Modelo Conceptual de Datos	96
Figura 4.20. Modelo de Servicios del Administrador	97
Figura 4.21. Modelo de Servicios del Coordinador de año.....	98
Figura 4.22. Modelo de Servicios del Director de Carrera.....	98
Figura 4.23. Modelo de Servicios del Estudiante.....	99
Figura 4.24. Modelo de Servicios del Docente	99
Figura 4.25. Modelo de Servicios Asistencia.....	99
Figura 4.26. Modelo de Servicios Calendario Académico	100
Figura 4.27. Modelo de Servicios del Horario de Clases.....	100
Figura 4.28. Modelo de Servicios del Horario de Clases.....	100
Figura 4.29. Modelo de Servicios del Horario de Clases.....	101

Figura 4.30. Modelo de Caso de Uso Extendido - Administrador	102
Figura 4.31. Modelo de Caso de Uso Extendido – Coordinador de Año	103
Figura 4.32. Modelo de Caso de Uso Extendido – Director de Carrera.....	104
Figura 4.33. Modelo de Caso de Uso Extendido – Docente	105
Figura 4.34. Modelo de Caso de Uso Extendido – Estudiante	106
Figura 4.35. Modelo de Caso de Uso Extendido - Asistencia.....	106
Figura 4.36. Modelo de Caso de Uso Extendido – Calendario Académico. ..	107
Figura 4.37. Modelo de Caso de Uso Extendido – Horario de Clases.....	107
Figura 4.38. Modelo de Caso de Uso Extendido – Justificación.....	108
Figura 4.39. Modelo de Caso de Uso Extendido – Reporte	109
Figura 4.40. Modelo de Composición de Servicio - Consultar Cabecera	110
Figura 4.41. Modelo de Composición de Servicio - Consultar Distribución ...	110
Figura 4.42. Modelo de Composición de Servicio - Consultar Estudiante	111
Figura 4.43. Modelo de Composición de Servicio - Guardar Asistencia.....	111
Figura 4.44. Modelo de Composición de Servicio - Ingresar Asistencia.....	112
Figura 4.45. Modelo de Composición de Servicio – Añadir Evento	113
Figura 4.46. Modelo de Composición de Servicio – Eliminar Evento	114
Figura 4.47. Modelo de Composición de Servicio – Consultar Eventos	115
Figura 4.48. Modelo de Composición de Servicio – Modificar Evento.....	115
Figura 4.49. Modelo de Composición de Servicio – Añadir Horario.	116
Figura 4.50. Modelo de Composición de Servicio – Consultar Horario.	116
Figura 4.51. Modelo de Composición de Servicio – Modificar Horario	117
Figura 4.52. Modelo de Composición de Servicio – Consultar Docente.....	118
Figura 4.53. Modelo de Composición de Servicio – Consultar Periodo.....	118
Figura 4.54. Modelo de Composición de Servicio – Iniciar Sesión.....	119
Figura 4.55. Modelo de Composición de Servicio – Validar Caracteres Especiales.....	119
Figura 4.56. Modelo de Composición de Servicio – Validar Campo Contraseña	120
Figura 4.57. Modelo de Composición de Servicio – Validar Espacio en Blanco	120
Figura 4.58. Modelo de Composición de Servicio – Validar Perfil Usuario	121
Figura 4.59. Modelo de Composición de Servicio – Consultar Asistencia.	121

Figura 4.60. Modelo de Composición de Servicio – Consultar Estudiante. ...	122
Figura 4.61. Modelo de Composición de Servicio – Ingresar Justificación	122
Figura 4.62. Modelo de Composición de Servicio –Modificar Asistencia.....	123
Figura 4.63. Modelo de Composición de Servicio – Generar Reporte.....	124
Figura 4.64. Modelo de Composición de Servicio –Agregar bitácora	124
Figura 4.65. Modelo de Composición de Servicio –Consultar bitácora	125
Figura 4.66. Modelo de Composición de Servicio –Agregar tiempo	125
Figura 4.67. Modelo de Composición de Servicio –Consultar tiempo	126
Figura 4.68. Modelo de Composición de Servicio –Agregar coordinador.....	126
Figura 4.69. Modelo de Composición de Servicio –Consultar coordinador. ..	127
Figura 4.70. Modelo de Composición de Servicio –Agregar Personalización.	127
Figura 4.71. Modelo de Fragmentos Estructurales.....	128
Figura 4.72. Modelo de Fragmentos Funcionales	129
Figura 4.73. Modelo de Navegación.....	130
Figura 4.74. Modelo de Servicio – Consultar Asistencia	131
Figura 4.75. Modelo de Servicio – Consulta Estudiante	131
Figura 4.76. Modelo de Servicio – Guardar Asistencia.....	132
Figura 4.77. Modelo de Servicio – Ingresar Asistencia.....	132
Figura 4.78. Modelo de Servicio – Añadir Evento.....	132
Figura 4.79. Modelo de Servicio – Eliminar Evento.	133
Figura 4.80. Modelo de Servicio – Consultar Evento.....	133
Figura 4.81. Modelo de Servicio – Modificar Evento.	133
Figura 4.82. Modelo de Servicio – Añadir Horario.	134
Figura 4.83. Modelo de Servicio – Consultar Horario.....	134
Figura 4.84. Modelo de Servicio – Modificar Horario.....	134
Figura 4.85. Modelo de Servicio – Consultar Docente.	135
Figura 4.86. Modelo de Servicio – Consultar Periodo.	135
Figura 4.87. Modelo de Servicio – Iniciar Sesión.....	135
Figura 4.88. Modelo de Servicio – Validar Caracteres Especiales.	136
Figura 4.89. Modelo de Servicio – Validar Caracteres Especiales.	136
Figura 4.90. Modelo de Servicio – Validar Perfil.....	136
Figura 4.91. Modelo de Servicio – Consultar Asistencia.	136

Figura 4.92. Modelo de Servicio – Consultar Estudiante.....	137
Figura 4.93. Modelo de Servicio – Ingresar Justificación.	137
Figura 4.94. Modelo de Servicio – Ingresar Justificación.	137
Figura 4.95. Modelo de Servicio – Ingresar Bitácora.....	138
Figura 4.96. Modelo de Servicio – Consulta Bitácora.....	138
Figura 4.97. . Modelo de Servicio – Ingresar Tiempo.....	138
Figura 4.98. Modelo de Servicio – Consultar Tiempo.....	139
Figura 4.99. Modelo de Servicio – Ingresar Coordinador	139
Figura 4.100. Modelo de Servicio – Consultar Coordinador.....	139
Figura 4.101. Modelo de Servicio – Modificar Opciones.	140
Figura 4.102. Modelo Objeto-Relacional.	141
Figura 4.103. Ejemplo de Código XML Schema para el fragmento consultarCabeceraAsistencia.....	142
Figura 4.104. Cabecera WSDL.....	142
Figura 4.105. Ejemplo de Código XML para los tipos de datos de los servicios.	142
Figura 4.106. Ejemplo de Código XML para los mensajes de las funciones.	143
Figura 4.107. Ejemplo de Código XML para las operaciones permitidas en los servicios.	143
Figura 4.108. Ejemplo de Código XML para especificación de protocolos de comunicación.	143
Figura 4.109. Ejemplo de Código XML correspondiente a los servicios Web.	144
Figura 4.110. Estructura de la capa de Datos.	144
Figura 4.111. Objeto de Clases del sistema.....	145
Figura 4.112. Objetos de Catálogos que se usaron en el sistema.	145
Figura 4.113. Formulario de bienvenida al sistema.	146
Figura 4.114. Detalle de asistencia mensual.....	147
Figura 4.115. Reporte de uso del sistema por parte de los docentes.....	148
Figura 116. Código XML Schema para el fragmento verificarPerfil.	162
Figura 117. Código XML Schema para el fragmento consultarDocente.....	162
Figura 118. Código XML Schema para el fragmento consultarTipoJustificacion	162

Figura 119. Código XML Schema para el fragmento consultarDistribucion ..	162
Figura 120. Código XML Schema para el fragmento consultarPeriodo	162
Figura 121. Código XML Schema para el fragmento consultarHorario	162
Figura 122. Código XML Schema para el fragmento consultarCabeceraAsistencia	163
Figura 123. Código XML Schema para el fragmento consultarJustificacion..	163
Figura 124. Código XML Schema para el fragmento consultarDetalleJustificacion	163
Figura 125. Código XML Schema para el fragmento consultarAsistencia.....	163
Figura 126. Código XML Schema para el fragmento consultarEstudiante	163
Figura 127. Código XML Schema para el fragmento consultarTipoAsistencia	164
Figura 128. Código XML Schema para el fragmento consultarEventos	164
Figura 129. Código XML Schema para el fragmento ingresarJustificacion ...	164
Figura 130. Código XML Schema para el fragmento modificarAsistencia.....	164
Figura 131. Código XML Schema para el fragmento ingresarAsistencia	164
Figura 132. Código XML Schema para el fragmento ingresarHorario	165
Figura 133. Código XML Schema para el fragmento modificarHorario	165
Figura 134. Código XML Schema para el fragmento ingresarEvento.....	165
Figura 135. Código XML Schema para el fragmento modificarEvento	165
Figura 136. Código XML Schema para el fragmento iniciarSesion	165
Figura 137. Código XML Schema para el fragmento consultarCoordinador .	166
Figura 138. Código XML Schema para el fragmento consultarTiempo	166
Figura 139. Código XML Schema para el fragmento consultarBitacora	166
Figura 140. Código XML Schema para el fragmento modificarPersonalizacion	166
Figura 141. Código XML Schema para el fragmento ingresarTiempo.....	167
Figura 142. Código XML Schema para el fragmento ingresarCoordinador. ...	167
Figura 143. Código XML Schema para el fragmento ingresarBitacora.....	167
Figura 144. Cabecera WSDL.....	169
Figura 145. Código XML para los tipos de datos de los servicios.	176
Figura 146. Código XML para los mensajes de las funciones.....	179

Figura 147. Código XML para especificación de protocolos de comunicación	187
Figura 148. Código XML correspondiente a los servicios Web	188

CUADROS

Cuadro 4.1. Definiciones, siglas y abreviaturas del ERS.....	73
Cuadro 4.2. Características de los computadores usados en las pruebas....	147
Cuadro 4.3. Comparación de Tiempo entre los procesos.	148

RESUMEN

El objetivo del presente trabajo fué desarrollar un módulo web de control de asistencia para los estudiantes de la Escuela Superior Politécnica Agropecuaria de Manabí ESPAM MFL, como requerimiento de la Comisión de Evaluación Interna. El mismo se lo realizo con el fin de optimizar los procesos de tiempos de entrega de los reportes mensuales, mejorando la calidad y presentación de los datos. La aplicación fue desarrollada en el lenguaje de programación C# con Visual Studio 2012 como IDE para la realización de la parte grafica que se mostrara al usuario y con el sistema gestor de base de datos SQL Server 2008 R2 para la manipulación de los datos y con la ayuda de varias herramientas informáticas más para un sistema óptimo. Se utilizó además la metodología ágil MIDAS para las distintas fases del desarrollo de la aplicación. Los resultados demuestran que la aplicación realizada optimizó los tiempos de entregas de reportes en un 99,80%, permitiendo con esto agilizar el proceso de análisis de los datos para los estudios de ausentismo de los estudiantes y verificar el cumplimiento de flujo de trabajo de los docentes.

PALABRAS CLAVES

Control de asistencia, Evaluación, Módulo web.

ABSTRACT

The objective of this work was to develop a attendance control web module for the Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, as a requirement of the Internal Evaluation Committee. It was performed in order to optimize the processes of delivering monthly reports, improving the quality and presentation of data. The application programming was developed in language C# with Visual Studio 2012 as IDE for the realization of the graphics that is displayed to the user and the SQL Server 2008 R2 management system database for handling data and various computer tools for optimal system. MIDAS agile methodology was also used for the various stages of application development. The results show an optimized delivery time of 99.80%, allowing it to streamline the process of data analysis for studies of student absenteeism and verify compliance with the workflow of teachers.

KEY WORDS

Attendance control, evaluation, web module.

CAPÍTULO I. ANTECEDENTES

1.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

Las universidades y escuelas politécnicas del Ecuador, son centros de formación de profesionales, comprometidos con la solución de problemas de producción y servicios mediante la investigación, por lo que el estado ecuatoriano debe garantizar que estos centros, aseguren la calidad de la educación. La Ley Orgánica de Educación Superior (2010), en su capítulo 2 art. 98-106, dispone una serie de requerimientos para evaluar a las universidades, y así certificar ante la sociedad, la calidad académica y la integridad institucional.

La Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López (ESPAM MFL) cuenta con un reglamento que regula los procesos de planificación, ejecución y evaluación de los resultados de aprendizaje de los cursos de la ESPAM MFL, para ejecutar este reglamento, existen una serie de indicadores que permiten evaluar el cumplimiento de los requerimientos, uno de estos indicadores es la evidencia de las hechos desarrollados en un periodo de estudio, demostrando el desempeño de los estudiantes, realizando una serie de actividades, estos son procesos manuales que podrían conllevar a la existencia de una brecha de errores humanos al momento de la recopilación de información, para evidenciar en el portafolio del curso. (ESPAM MFL, s.f.)

Unas de las actividades que forma parte de este portafolio es el registro de asistencia de los estudiantes, el mismo que se debe entregar de manera mensual, informes impresos, sin embargo, no todos los docentes cumplen con esta disposición puntualmente, por el lado contrario, entregan la totalidad de los datos al final de cada semestre, plasmando información no precisa en los reportes presentados por los mismos, y a la vez distorsionando el conjunto de hechos que conforman las evidencias del cumplimiento de los logros de aprendizaje de los cursos.

El departamento de acreditación y evaluación de la universidad es el encargado de llevar a cabo todos los trámites que confieren a la acreditación de la universidad y además, es quien fiscaliza que los procesos académicos que se están llevando a cabo en los predios de la institución sean los más adecuados para poder, en primera instancia, ofrecer una buena calidad de educación a los alumnos en un ambiente acorde a los estatutos que se exigen, y en segunda instancia, para lograr obtener una buena calificación ante el ente de evaluación que es el Consejo de Evaluación Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAASES), y de esta manera poder acreditarse, razón por la cual, el Departamento de Acreditación y Evaluación, tiene la necesidad de un módulo web que gestione el proceso de entrega de reportes a tiempo por parte de los docentes y así poder llevar el control de las asistencias de los estudiantes eficientemente, siendo este uno de los ítems del portafolio que sirve para realizar una evaluación a los docentes.

Por los motivos anteriormente formulados, los autores se plantearon la siguiente interrogante:

¿Cómo lograr la optimización de los procesos de presentación de reportes del control de asistencia de los estudiantes de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López de la ciudad de Calceta?

1.2. JUSTIFICACIÓN

En una época en donde la tecnología predomina, los sistemas informáticos son cada vez más indispensables debido a la gran cantidad de información en las instituciones, y la implementación de estos aplicativos permitirá manejar dicha información de manera ágil, lo que es de gran beneficio para las organizaciones poder contar con un sistema de información mediante el cual se pueda recopilar estos datos de manera organizada para un acceso rápido a ellos, y lograr analizarlos de manera eficaz.

Para la ESPAM MFL la implementación de un módulo web para el control de asistencia beneficiaría y a la vez aportaría positivamente al sistema integral de la universidad, además, ayudaría a reducir la cantidad de papel que se emplea en los reportes impresos de asistencias, beneficiando a futuro al medio ambiente, ya que también se lograría reducir el costo económico que se presenta al momento de adquirirlo para la impresión de estos reportes.

La ejecución de este proyecto, busca mejorar los procesos de control de asistencia de los cursos de la ESPAM MFL, llegando a optimizar los tiempos de presentación de reportes, ayudando a los docentes y estudiantes a obtener un mejor control de sus propias asistencias a los distintos cursos que están tomando en los predios de la universidad.

La presente propuesta también aporta a la automatización de una de las actividades, que se tienen que realizar para el cumplimiento del portafolio de evidencias, y así optimizar procesos, logrando de esta manera, realizar un aporte al sistema integral de evaluación de las carreras de la universidad, ejecutando cada uno de estos de manera eficiente, y poder alcanzar una buena calificación en los procesos de evaluación de la universidad mediante el cumplimiento de los requerimientos de la Ley Orgánica de Educación Superior (2010) que están contemplado en el art 151 y 155. Con la automatización de estas actividades los docentes serán evaluados verificando el cumplimiento de la actividad de control de asistencia respectivo.

Por lo antes mencionado, se realizó la ejecución de este proyecto y se consiguió el cumplimiento del reglamento de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, como lo establece en el Manual de Sistema de Investigación Institucional, referente al reglamento de Tesis de Grado en su Art. 2 indica “Todo tema de tesis estará relacionado con las líneas de investigación de la carrera del postulante, enmarcadas en las áreas y prioridades de investigación establecidas por la ESPAM-MFL”.

1.3. OBJETIVOS

1.3.1 OBJETIVO GENERAL

Elaborar un módulo web para el control de asistencia de los estudiantes de pregrado, que optimice los procesos de presentación de reportes en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

1.3.2 OBJETIVOS ESPECÍFICOS

- Analizar la información recopilada para los esquemas del modelo independiente de computación necesarios para el desarrollo del sistema.
- Definir los modelos independientes de la plataforma, especificando formalmente el sistema.
- Estructurar los modelos específicos de plataforma, describiendo los detalles para implementar en la plataforma.
- Construir el módulo web en base al diseño realizado previamente.
- Verificar el óptimo funcionamiento del módulo desde la página principal de la universidad.

1.4. HIPÓTESIS, PREMISA Y/O IDEAS A DEFENDER

1.4.1. IDEAS A DEFENDER

La elaboración de un módulo web para el control de asistencia de los estudiantes de pregrado, optimizará el proceso de presentación de reportes en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

CAPÍTULO II. MARCO TEÓRICO

2.1. SISTEMA DE EDUCACIÓN SUPERIOR EDUCATIVO

La Asamblea Constituyente (2008) fundamenta en el Art. 350 que el Sistema de Educación Superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo.

A su vez Asamblea Constituyente (2008) en el Art. 351 aclara que este sistema regirá por los principios de autonomía responsable, cogobierno, igualdad de oportunidades, calidad, pertinencia, integralidad, autodeterminación para la producción del pensamiento y conocimiento, en el marco del diálogo de saberes, pensamiento universal y producción científica tecnológica global.

Además determina Asamblea Constituyente (2008) en el Art. 352 que el Sistema de Educación Superior estará integrado por universidades y escuelas politécnicas; institutos superiores técnicos, tecnológicos y pedagógicos; y conservatorios superiores de música y artes, debidamente acreditados y evaluados. Estas instituciones, sean públicas o particulares, no tendrán fines de lucro.

Dado los conceptos los autores definen los artículos de la Constitución Ecuatoriana como fuente de conocimiento para el presente trabajo.

2.1.1. UNIVERSIDADES Y ESCUELAS POLITÉCNICAS

Según el Poveda et al. (1994) las Universidades y Escuelas Politécnicas son comunidades de profesores, estudiantes y trabajadores que buscan la verdad y el desarrollo de la ciencia y la cultura, mediante la docencia y la investigación. Están abiertas a todas las corrientes del pensamiento universal.

Según los autores, universidad es una institución académica la cual se dedica a la enseñanza académica en un nivel superior, en que sus carreras están

enfocadas a temas humanísticos, es decir, carreras relacionadas la filosofía, biología, derecho, entre otros; incentivando a la investigación, la misma está autorizada para otorgar títulos profesionales luego de haber cursado el tiempo correspondiente por cada área de estudio.

Los autores definen como Escuela Politécnica a los institutos de educación superior, estas ofrecen un gran variedad de carreras que pueden contemplar asignaturas como física, química, informática, a la vez complementar con otras, como pueden ser geología, eléctrica, mecánica, pero en su gran mayoría están orientadas a las tecnologías, el nivel de investigación responde a la necesidad de la resolución de problemas de la sociedad, la formación de los estudiantes se realiza de manera integral para obtener a una vida productiva.

Por lo que los representantes de este trabajo asumen que las instituciones académicas de Educación Superior están enfocadas a desarrollar el área social, económica, política, y cultural del país.

2.2. PROCESO

En la definición de González (2006) visto desde el lado industrial un proceso es una serie de actividades que conducen a un cierto producto, el cual es el objetivo de dicho proceso. Sin embargo la definición de proceso desde el punto de vista de la calidad un proceso es una secuencia de actividades cuyo producto crea un valor para el usuario o cliente.

La norma UNE-EN ISO 9000:2000 por otro lado define al proceso como un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

Por su parte la Junta de Castilla y León (2006) define proceso en el ámbito administrativo como la secuencia ordenada de actividades, incluidos los trámites de los procedimientos administrativos, interrelacionadas entre sí, precisas para dar respuesta o prestar servicio al ciudadano, como cliente, usuario o beneficiario de servicios o prestaciones.

Dado los conceptos los autores apoyan su trabajo de tesis en la definición de Junta de Castilla y León (2006), ya que se apega en la forma en que funciona el sistema.

2.3. OPTIMIZACIÓN

Según Duarte (2007) la optimización es una disciplina fundamental en campos de la ciencia tales como la informática, la inteligencia artificial o la investigación operativa. En otras comunidades científicas la definición de optimización se torna bastante impreciso. Se concibe también como el proceso de intentar encontrar la mejor solución posible a un problema de optimización, generalmente en un tiempo limitado.

2.3.1. OPTIMIZACIÓN DE APLICACIONES

Optimizar una aplicación significa hacer los cambios pertinentes para que esta se ejecute y funcione más rápidamente, o para que ocupe menos memoria, o para que gaste menos batería (útil en computadoras portátiles). Por lo general la optimización de un programa se hace a través de otros programas, una mejor configuración o similares, pero siempre a nivel software.

De todas maneras, la mejor optimización que puede hacerse a una aplicación es a nivel código fuente, es decir, cambios en las sentencias de programación. Mejorar los algoritmos resulta en una mejora del rendimiento general de la aplicación. También puede emplearse un compilador optimizador que ayude a crear una aplicación más optimizada. (Alegsa, 2010c)

Los autores harán de uso de la definición de Alegsa (2010c), para que el proyecto esté sustentado en la teoría que los procesos que son optimizados serán de gran utilidad en la colectividad involucrada.

2.4. CONTROL DE GESTIÓN

El control dado la afirmación de Pérez y Juan (2006) «es aquella función que pretende asegurar la consecución de los objetivos y planes prefijados en la fase de planificación. Como última etapa formal del proceso de gestión, el

control se centra en actuar para que los resultados generados en las fases anteriores sean los deseados».

Se entiende Ponce (2010?) por control de gestión al conjunto de procesos que la empresa aplica para asegurarse de que las tareas que en ella se realizan están encaminadas a la consecución de los objetivos previamente establecidos en el proceso de planeamiento estratégico.

El control de gestión no se limita a comprobar que las tareas realizadas o las decisiones tomadas han sido correctas, sino que parte de su cometido es influir y orientar el comportamiento de la organización para que se alcancen los objetivos propuestos.

Según los conceptos los autores incluirán las definiciones de Ponce (2010?) ya que ofrece definiciones claras del control de gestión.

2.4.1. CONTROL DE ASISTENCIA

El control de asistencia del alumnado al aula es una tarea que el maestro o la maestra realiza diariamente, tanto para un seguimiento individual de las vicisitudes por las que pasa cada alumno, como para saber con quién cuenta para la organización de las actividades del día. Para llevar este control, se usa necesariamente la lista escrita de los nombres de los estudiantes.

Por otro lado, también conviene comentar con todo el grupo-clase, antes de comenzar las tareas del día. Si falta alguien, si alguien que estaba enfermo ha vuelto, si hay algún cumpleaños, etc. a fin de crear un clima afectivo que cohesione el grupo y permita un marco emocionalmente estable para desarrollar tareas intelectuales.

Según la organización de las actividades de la clase, surgen también otras necesidades de controles de asistencia o similares. Así, por ejemplo, en los ratos de trabajo en que el alumnado se distribuye en diferentes actividades (rincones, talleres, grupos de trabajo, etc.) surge la necesidad de llevar un cierto control de la distribución, tanto si es elegida libremente como si es asignada, para posibilitar la alternancia de actividades.

Así pues, vemos que para un buen funcionamiento de la dinámica del grupo-clase, el maestro comparte con el alumnado diversos controles de asistencia. Este hecho se canaliza necesariamente a través de la lengua escrita, con la coincidencia de que para esta actividad, hay que leer los nombres de los miembros de la clase, lo cual resulta muy útil. (Fons, 2006.).

2.5. PORTAFOLIO EDUCATIVO

Recalca García (2005), a pesar de que el portafolio no surge en ámbitos educativos, se ha utilizado desde siempre, incluso algunos docentes sostienen que los portafolios no representan una novedad en la educación, porque tradicionalmente el docente siempre ha realizado una recopilación de trabajos.

La apreciación de Hilda (1996) habla que el uso del portafolio radica en la necesidad de realizar una planificación sistemática del proceso de aprendizaje en el aula, de modo que recoja el trabajo real que los estudiantes llevan a cabo durante el proceso. En relación con Barragán (2005) afirma que el portafolio se convierte en una metodología de enseñanza y evaluación que hace su aparición como metodología alternativa a aquellas de corte puramente cuantitativo, se trata realmente de un procedimiento de evaluación de trayectorias de aprendizaje que se basa en las ejecuciones y logros obtenidos por los participantes en dichas trayectorias y que además incorpora el valor añadido de su potencial de aprendizaje.

En este sentido Prendes y Sánchez (2008) reflexionan que el portafolio es una compilación de trabajos que realiza un alumno en relación con unos objetivos específicos predefinidos, trabajos a los que se unen comentarios y reflexiones tanto por parte del propio alumno como del profesor. Desde la perspectiva del profesor, un portafolio es una recopilación de tareas, anotaciones, sucesos y reflexiones acerca de un proceso de enseñanza. A partir de esta idea, se entiende el portafolio educativo como una metodología de enseñanza que implica algo más que la mera recogida de trabajos, el portafolio educativo implica una reflexión, una recogida de experiencias, que permite acercar su realidad a la persona destinataria del portafolio, permite analizar, valorar, revisar, evaluar,....

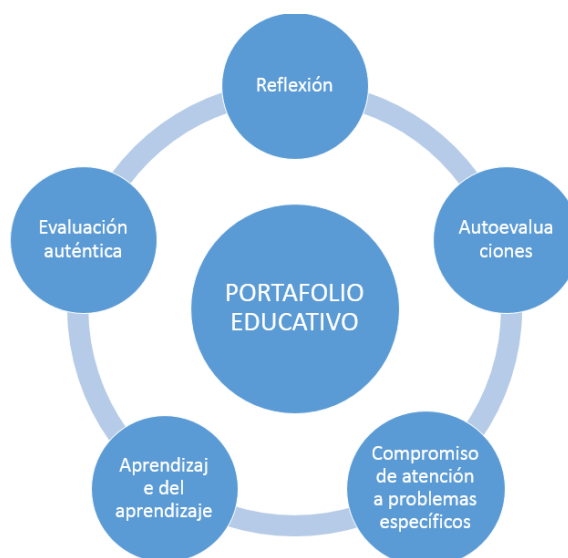


Figura 2.1. Acciones que implica el portafolio electrónico

2.5.1. PORTAFOLIO DEL PROFESOR

El portafolio ofrece al profesor una reflexión crítica de su propia actividad docente, y puede cumplir, además, una interesante función como herramienta para la evaluación de pares, al colaborar con otros docentes en el intercambio de portafolios se conoce la trayectoria desarrollada por los profesores y se favorece la comunicación de experiencias. Se pueden incluir además documentos de alumnos, calificaciones, trabajos, etc.... (Prendes y Sánchez, 2008).

2.5.2. PORTAFOLIO ELECTRÓNICO

Prendes y Sánchez (2008) refiere que puede ser llamado webfolio, portafolio digital y portafolio multimedia. “Contiene la misma información que un portafolio educativo tradicional pero en este caso el material es presentado en formato digital [...] utilizando una combinación de tecnologías multimedia”.

Así, estos portafolios pueden incluir grabaciones de audio, imágenes, programas informáticos, bases de datos, vídeos, páginas web,... y pueden ser presentados a través de distintos soportes informáticos o incluso a través de la red Internet. (MyLabSchool, 2007).

Los resultados de Prendes y Sánchez (2008) demuestran que portafolio electrónico presenta las características del portafolio tradicional y de forma

añadida todas las posibilidades de un documento flexible y susceptible de continuos cambios. Puede además estar almacenado en un soporte físico (CD, DVD,...) o bien puede ser desarrollado en red. Para posibilitar esta flexibilidad existe toda una gama de herramientas generales y específicas que gracias a Internet ven incrementadas sus posibilidades de comunicación.

2.6. INTERNET

2.6.1. CONCEPTOS BÁSICOS DE INTERNET

Para Ramos y Ramos (2011) el Internet es una gran red mundial de ordenadores conectados que como todas las redes, permite compartir recursos e intercambiar información. Podemos comunicarnos con personas ubicadas en diferentes partes del mundo, ya sea por motivos personales, de trabajo, académicos o simplemente de ocio. En esta gran red mundial, los ordenadores están unidos a través de conexiones de varios tipos y para comunicarse utilizan un lenguaje o protocolo común, el TCP/IP.



Figura 2.2. Red mundial de ordenadores

2.6.2. FUNDAMENTOS DE LA WEB

El punto de vista de Ramos y Ramos (2011) que los conceptos de internet y web a menudo se confunden y se tienden a creer que son lo mismo, pero en realidad son diferentes. Internet engloba a las tecnologías que hacen posible que los ordenadores ubicados en diferentes lugares del mundo se conecten entre sí y puedan compartir información. Esta conexión es posible mediante

cables, módems, líneas telefónicas, routers, protocolos, entre otros. Concluyendo que todo ese conjunto de tecnologías se le llama internet. La web es solo una de los servicios que proporciona Internet.

La web fue creada en 1989 por Tim Bérners Lee. Consistía en una forma de organizar la información usando como medio físico la comunicación de la red Internet y el protocolo HTTP. HTTP (Hypertext Transference Protocol) es el protocolo de transferencia de hipertexto que lo navegadores utilizan para realizar las peticiones a los servidores web y para recibir respuestas de ellos. Es el protocolo que se utiliza para ver las páginas web. (Ramos y Ramos, 2011).

2.6.3. WEB 2.0

Van (2005) puntualiza que la Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final. De igual manera afirma es una actitud y no precisamente una tecnología.

También Van (2005) explica que es una transición de aplicaciones tradicionales hacia aplicaciones que funcionan a través de las web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio.

Al profundizar sobre este tema Van (2005) considera que es una etapa que ha definido nuevos proyectos en Internet y está preocupándose por brindar mejores soluciones para el usuario final. Muchos aseguran que se ha reinventado lo que era el Internet, otros hablan de burbujas e inversiones, pero la realidad es que la evolución natural del medio realmente ha propuesto cosas más interesantes, afirmando que en el momento que la web inició, se encontraba en un entorno estático, con páginas en HTML que sufrían pocas actualizaciones y no tenían interacción con el usuario.

2.6.4. NAVEGADOR WEB

Un navegador web (del inglés web browser) es un programa que permite visualizar la información que contiene una página web, que interpreta el código de la página (normalmente en HTML) y lo visualiza en la pantalla. Es la

herramienta que utiliza el usuario para moverse e interactuar con la información contenida en las distintas páginas web de la red. (Ramos y Ramos, 2011).

2.7. SERVIDORES WEB

Un servidor web es un programa diseñado para aceptar peticiones HTTP del navegador y servir las páginas web que tiene alojadas. Para navegar a un sitio web se necesita saber la dirección web exacta del sitio. Cada sitio web tiene una dirección única conocida como URL Localizador Uniforme de Recursos). En una URL se pueden distinguir tres partes: el protocolo, el nombre de dominio del ordenador servidor y la ruta en el servidor. El protocolo que se utiliza para servir páginas web es el HTTP, el nombre de dominio indica el ordenador de Internet que nos va a dar la información solicitada y la ruta indica la ubicación del archivo solicitado dentro del servidor. (Ramos y Ramos, 2011).

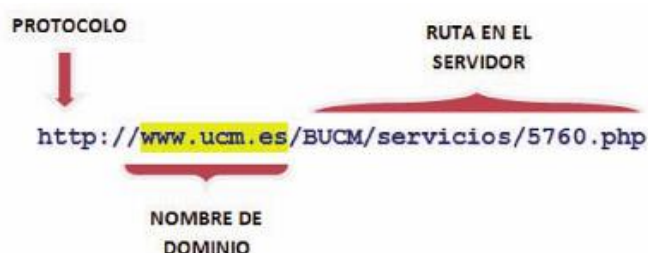


Figura 2.3. Estructura de una dirección web

2.7.1. DIRECCIONES IP O DIRECCIONES INTERNET

Son las direcciones que utiliza el protocolo IP (Internet Protocol - Protocolo de Internet) para identificar de forma única un dispositivo en Internet. Este protocolo permite que se comuniquen los dispositivos conectados a la red, ya sean ordenadores, routers, puntos de acceso, dispositivos móviles, servidores, etc. Una dirección IP es un código numérico que posee cuatro valores separados por puntos, y que pueden variar de 0 a 255 (en total. 256 valores posibles). Algunos ejemplos de direcciones IP son:

-172.16.1.12 (correcto)

- 77.225.48.6 (correcto)

- 345.262.1.1 (incorrecto, los dos primeros valores son mayores que 255). (Ramos y Ramos, 2011).

2.7.2. DOMINIOS

Son los nombres de los sitios web que son visitados. En lugar de utilizar las direcciones IP para acceder a los sitios web, se utilizan los nombres de dominio, ya que es más sencillo recordar el nombre de un sitio que su dirección IP. Se pueden clasificar en dos tipos:

- a) Internacionales o genéricos: hacen referencia al tipo de organización que es dueña del dominio. Algunos ejemplos son: .com (compañía comercial), .org (organización sin fines de lucro), .net (redes y servicios de Internet), .edu (institución educativa), .gob (organización gubernamental).
- b) Dominios territoriales o dominios país: representan a un país y están a cargo del estado de cada nación. Por ejemplo: .es (España), .fr (Francia), .uk (Reino Unido), .ar (Argentina), .de (Alemania). (Ramos y Ramos, 2011).

2.7.3. PROVEEDORES DE SERVICIOS DE INTERNET, ISP (INTERNET SERVICE PROVIDER)

Para establecer una conexión a internet se necesita un proveedor de servicios de Internet o ISP. Son empresas dedicadas a conectar a Internet a los usuarios o las distintas redes que tengan, y a hacer el mantenimiento necesario para que el acceso funcione correctamente. También ofrecen servicios relacionados, como alojamiento web o registro de dominios, entre otros.

Un ISP es como una compañía proveedora de servicios telefónicos. Se paga por meses, o a veces por una determinada cantidad de horas; o bien se paga para tener acceso ilimitado. El coste varía en función del plan que se elija. Este servicio generalmente incluye una cuenta de correo electrónico gratis y a veces una página de Internet. Son proveedores de servicios Internet empresas como Jazztel, Ono, Vodafone, Movistar y Orange, entre otras. (Ramos y Ramos, 2011).

2.7.4. SERVIDORES DNS (DOMAIN NAME SERVER - SERVIDOR DE NOMBRES DE DOMINIO)

Son los que resuelven el nombre de un dominio en una dirección IP determinada (así no es necesario tener que recordar y usar la dirección IP). Cada servidor mantiene una base de datos de nombres y direcciones para el dominio local. Para satisfacer las solicitudes fuera del dominio local, cada servidor almacenará también los nombres y direcciones de otros servidores de nombres. Todo este proceso de resolución de nombres es transparente para el usuario.

De una forma sencilla, el funcionamiento es el siguiente: cuando se desea localizar, por ejemplo, una página web, se escribe `www.google.es` en la barra del navegador. En primer lugar, el navegador identifica la URL y, a continuación, solicita al servidor DNS la dirección IP. El servidor e indica al navegador que tiene que ir a buscar la información de la página web a la IP `74.125.230.81`. (Ramos y Ramos, 2011)



Figura 2.4. Búsqueda de la IP en el servidor DNS

El navegador establece una conexión con la dirección IP proporcionada por el servidor DNS. El servidor web envía el archivo correspondiente y el navegador muestra la página solicitada. Todo esto ocurre en milésimas de segundo. (Ramos y Ramos, 2011)



Figura 2.5. Devolución de la web solicitada

2.8. CLIENTE

Luján (2002) señala que el cliente web es un programa con el que interacciona un usuario para solicitar a un servidor web, el envío de los recursos que desea obtener mediante HTTP. En otro aspecto la parte cliente de las aplicaciones web suele estar formada por el código (HTML) que forma la página web más algo del código ejecutable realizado en el lenguaje script del navegador (JavaScript o VBScript) o mediante pequeños programas. También se suelen emplear plug-ins que permiten visualizar otros contenidos multimedia (como Flash) aunque no se encuentran tan extendidos como las tecnologías anteriores y plantean problemas de incompatibilidad entre las diferentes plataformas. Por lo tanto, la del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen.

2.9. APLICACIÓN WEB

Una aplicación web se puede representar en 3 capas: la capa superior (el cliente web, normalmente un navegador), el nivel intermedio que procesa los datos (el servidor web) y el nivel inferior que proporciona los datos (la base de datos). (Luján, 2002).

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones (TCP/IP), que son los empleados en internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre los distintos ordenadores. (Luján, 2002).

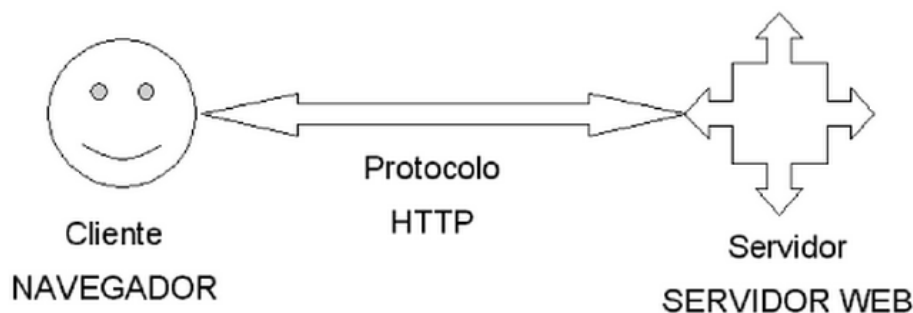


Figura 2.6. Esquema básico de una aplicación web

En términos prácticos, las aplicaciones web son soluciones informáticas o como su nombre lo dice “aplicaciones” que permiten interactuar con la información y a las cuales se puede acceder a través de una conexión a internet, desde cualquier lugar del mundo, sin necesidad de instalarlas previamente en nuestra pc; solamente se necesita contar con un navegador web, llámese Mozilla, Internet Explorer, Chrome, entre otros. (E-lemental, 2010).

2.10. MÓDULO WEB

En programación, un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo. Los módulos promueven la modularidad y el encapsulamiento, pudiendo generar programas complejos de fácil comprensión. (Alegsa, 2010a).

Según Rocha, (2014) un módulo es un subprograma, el cual ha sido desarrollado para llevar a cabo una tarea específica. La estructura de un programa modular, los módulos dependen de un módulo principal, desde el cual son llamados.

Según la definición los autores concluyen, un módulo web es un subprograma que está integrado a un sistema web.

2.11. SERVICIO

En informática, un servicio es un procedimiento, un método o un objeto con una interfaz estable y pública que puede ser invocado por un cliente, en cualquier instancia del sistema. En cambio los Servicios Web amplían esa idea para permitir que esa invocación se realice a través de internet empleando protocolos Web estándar ya existentes. (CCIA, 2008)

Se puede afirmar según el concepto, que servicios son métodos que están disponibles en el sistema para ser usados por el cliente en cualquier momento de la ejecución.

2.11.1. SERVICIOS WEB

Un Servicio Web para CCIA, (2008) es un componente software que puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. Permiten exponer y hacer disponibles funcionalidades (servicios) de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos WEB estándar. Cada Servicio Web tiene la responsabilidad de realizar un conjunto de funciones concretas y bien definidas Servicios Web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

En la investigación de Alegsa (2010b) concluye que la W3C define Servicio web como un sistema de software diseñado para permitir interoperabilidad máquina a máquina en una red. En general, los servicios web son sólo APIs Web que pueden ser accedidas en una red, como internet, y ejecutadas en un sistema de hosting remoto. En términos sencillos, un servicio web es cualquier sistema de software diseñado para soportar interacción máquina a máquina sobre una red. Esta amplia definición abarca múltiples y diferentes sistemas, pero en general servicio web suele referir a clientes y servidores que se comunican usando mensajes XML que siguen el estándar.

Las definiciones de servicios web sustentan el presente trabajo, dado la afirmación, que permite la interacción entre el cliente y el servidor.

2.11.1.2. JSON

JSON (JavaScript Object Notation) para es un formato utilizado en JavaScript (JS) para señalar datos. Se lo suele usar en sitios web como formato de intercambio de datos "liviano" (en comparación con XML), y tiene la particularidad que los datos en este formato son leídos de manera directa en JavaScript. Si bien la manera "natural" de señalar datos con Python es usando pickle, igual los programadores Python suelen usar JSON para evitar problemas de seguridad. (Bassi, S. 2013).

Para Esquivá (2013) señala que es un formato para el intercambios de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. JSON nació como una alternativa a XML, el fácil uso en JavaScript ha generado un gran número de seguidores de esta alternativa. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.

Los autores asumen que JSON es una sintaxis para el intercambio de datos que puede ser utilizado en cualquier lenguaje de programación.

2.11.1.2. SIMPLE OBJECT ACCESS PROTOCOL (SOAP)

SOAP, según la definición de Sudha y Sujatha (2011) originalmente definida como Simple Object Access Protocol, es una especificación de protocolo para el intercambio de información estructurada en la implantación de servicios web en redes informáticas. Se basa en XML para el formato del mensaje, y por lo general se basa en otros protocolos de capa de aplicación, llamada a procedimiento más notablemente remoto (RPC) y el protocolo de transferencia de hipertexto (HTTP), para la negociación y transmisión de mensajes. SOAP puede formar la capa de base de una pila de protocolos de servicios web, proporcionando un marco de mensajería básica sobre la que los servicios Web se pueden construir. Este protocolo basado en XML se compone de tres partes: un sobre, que define lo que está en el mensaje y cómo procesarlo, un conjunto de reglas de codificación para expresar instancias de tipos de datos definidos por la aplicación, y una convención para representar llamadas a procedimientos y respuestas.

Según Montilva et al. (2011), es un protocolo de comunicación, por paso de mensajes XML, se utiliza para la formación de los mensajes intercambiados entre los sistemas distribuidos y la red. Los mensajes SOAP son independientes de los sistemas operativos y de los protocolos, y pueden ser transportados usando una variedad de protocolos Internet, incluyendo SMTP, y HTTP. El mensaje está compuesto de tres partes: un sobre, un encabezado y el cuerpo.

Los autores sustentan el presente trabajo en la definición de Montilva et al. 2011, en que apoya que es un protocolo de intercambio de información entre el servidor, y el aplicativo web.

2.12. BOOTSTRAP

Fontela (2013).precisa que este framework que permite crear interfaces web con CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de cualquier dispositivo sin intervención del usuario, esto se denomina diseño adaptable o Responsive Design, un ejemplos de la implementación de esta tecnología es claramente twitter.

Por otro lado para Gutiérrez (2012) es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por Twitter. La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (Responsive Design) apoyándonos en un framework potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo.

Los autores sustentaran su diseño del adaptable en este framework, con la ayuda de estos conceptos.

2.12.1. CARACTERÍSTICAS PRINCIPALES DE BOOTSTRAP

Bootstrap ofrece una serie de plantillas CSS y ficheros JavaScript que nos permiten integrar el framework de forma sencilla y potente en nuestros proyectos webs.

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo JQuery.
- Ofrece un diseño sólido usando (lenguaje de hojas de estilo) LESS y estándares como CSS3/HTML 5.
- Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los tags HTML5.
- Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos. (Gutiérrez, 2012)

2.13. REPORTE

Los autores describen que un reporte es un documento en el cual se detalla de una manera resumida todas las actividades de mayor relevancia para una empresa, en ellas se podrían encontrar datos estadísticos sobre el control de cada uno de los procesos por los cuales está constituido una organización, además de tener una visión mucho más esquematizada sobre el rendimiento de la misma. Estos son generalmente presentados mediante algún tipo de software y agrupan los datos de acuerdo a un interés en específico. Por ejemplo si se deseara presentar un reporte sobre el control de las asistencias que tienen los alumnos a alguna materia de clases, se presentaría toda la lista de estudiantes con su porcentaje de asistencia respectiva a cada materia de clases.

2.14. PRESENTACIÓN

El objetivo principal según los autores, una presentación es brindar ideas, dar a conocer información o conocimiento que se necesita, se entienda, de una manera mucho más sencilla. Es un gran instrumento de trabajo para las

organizaciones cuando las mismas necesiten expresar información con respecto a cómo se encuentran los procesos de la organización.

2.14.1. PRESENTACIÓN DE REPORTE

La presentación de reportes según los autores, es la manera en cómo los datos finales se presentan a la persona interesada en recibir el reporte, se puede hacer una presentación ya sea de manera escrita en un papel o digital mediante la web o algún tipo de software que facilite la creación de los mismo, o también en las mil y un manera que pueden existir para poder presentar un reporte. Estos tienen que ser representativos, llamativos y entendibles para la mayoría de las personas que se encuentren ligados al entorno en el que se está elaborando.

2.15. BASES DE DATOS

Según Camps et al. (2005) definen que las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron.

Desde la realización del primer modelo de datos, pasando por la administración del sistema gestor, hasta llegar al desarrollo de la aplicación, los conceptos y la tecnología asociados son muchos y muy heterogéneos. Sin embargo, es imprescindible conocer los aspectos clave de cada uno de estos temas para tener éxito en cualquier proyecto que implique trabajar con bases de datos. (Camps *et al.*, 2005).

2.15.1. BASE DE DATOS

Gómez y Martínez (1997) define una base de datos es un conjunto de información en forma normalizada, almacenada en cualquier dispositivo y entre la que se establece un número limitado o ilimitado de relaciones. Esta

información puede ser de muchas clases: desde registros de ventas de un negocio, a una lista de invitados a un restaurante.

Pérez (2007) manifiesta que una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consisten en tres campos: nombre, dirección, y número de teléfono.

Mientras que para Cobo (2008) una base de datos es un conjunto de datos almacenados sin redundancias innecesarias en un soporte informático y accesible simultáneamente por varios usuarios y aplicaciones. Los datos deben de estar estructurados y almacenados en forma totalmente independiente de las aplicaciones que la utilizan.

2.15.2. SISTEMA GESTOR DE BASES DE DATOS

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos

van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos (Silberschatz *et al.*, 2002).

Para Cobo (2008) un sistema gestor de bases de datos es un software o conjunto de programas que permite crear y mantener una base de datos. El objetivo principal de un SGBD es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos.

Este software facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones.

2.16. LENGUAJES DE PROGRAMACIÓN PARA LA WEB

Existen números lenguajes de programación para el desarrollo de aplicaciones web, y son ejecutados tanto del lado del cliente como del servidor, entre los que destacan como lenguajes del lado del cliente se encuentran:

- ✓ HTML
- ✓ Java con sus tecnologías Java Servlets y JavaServer Pages (JSP)

Y por otro lado se encuentran los lenguajes que se ejecutan del lado del servidor, entre los cuales destacan:

- ✓ PHP
- ✓ ASP/ASP.NET
- ✓ Perl
- ✓ Ruby
- ✓ Python

También existen tecnologías que permiten ejecutarse tanto del lado del cliente como del servidor, por ejemplo:

- ✓ XML

ASP es una arquitectura de desarrollo Web en los que se puede usar por debajo en distintos lenguajes (por ejemplo VB.NET o C# para ASP.NET o VBScript/JScript para ASP). (Caivano y Villoria, 2009).

2.16.1. ASP

(Alvarez, 2001) resalta que ASP es una tecnología desarrollada por Microsoft para la creación de páginas dinámicas. La cual se escribir en la misma página

web, utilizando el lenguaje Visual Basic Scripto Jscript (JavaScript de Microsoft).

(Alvarez, 2001) opina que un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente, por lo que permite realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP, la cual es compatible con todos los navegadores.

Alvarez (2001) concluyendo que en las páginas ASP se pueden realizar muchos tipos de aplicaciones distintas desde acceso a bases de datos, al sistema de archivos del servidor hasta admite el uso de los recursos que tenga el propio servidor.

Para Navarro et al., (2006) ASP es una tecnología desarrollada por Microsoft para crear páginas web de contenidos dinámicos apoyándose en scripts ejecutados en el servidor. Básicamente una página ASP es una mezcla entre una página HTML y un programa que da como resultado una página HTML que es enviada al cliente (navegador). ASP es una tecnología que pertenece a la parte servidor por eso no es necesario que el cliente o navegador lo soporte ya que se ejecuta en el servidor, sí que deberemos buscar un servidor que soporte este tipo de tecnología para que las páginas se ejecuten correctamente.

2.16.2. ASP.NET

ASP.NET es un marco de trabajo web que le ayuda a crear soluciones web basadas en estándares. Proporciona un avanzado conjunto de características preparadas para su uso y controla y facilita la integración de bases de datos, servicios web y aplicaciones de Internet avanzadas. Con características como la técnica de scaffolding de datos dinámica, un marco de trabajo AJAX, MVC, temas y plantillas, cientos de controles integrados y herramientas de desarrollo gratuitas, ASP.NET le permitirá crear aplicaciones web mejores y más rápidas. (Microsoft, 2007b).

2.16.3. INTRODUCCIÓN A VISUAL STUDIO

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y logra hacer más sencilla la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. (Microsoft, 2007a).

2.17. METODOLOGÍA DE DESARROLLO MIDAS

En la investigación de Zorrilla y Vela (2007) es una metodología dirigida por modelos para el desarrollo de Sistemas de Información Web (SIW), que propone el uso de estándares a lo largo de todo el proceso de desarrollo, así como el uso de UML para el modelado del SIW independientemente del nivel de abstracción o del aspecto del sistema a modelar.

En el análisis de Zorrilla y Vela (2007) afirman que dado que UML no permite representar directamente todos los modelos necesarios, MIDAS incorpora algunas extensiones de UML existentes y define, o adapta, otras nuevas, siempre que es necesario. MIDAS propone una arquitectura dirigida por modelos y considera a la hora de modelar el sistema los aspectos de contenido, hipertexto y comportamiento.

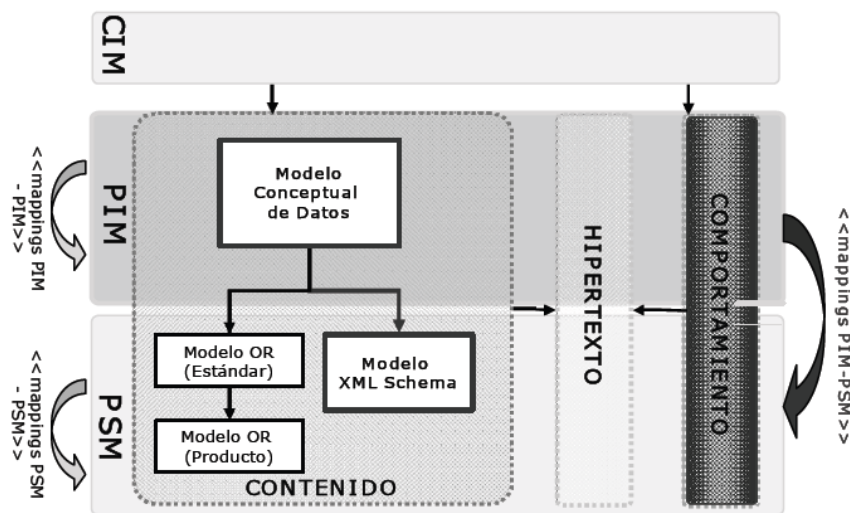


Figura 2.7. Arquitectura de MIDAS simplificada

Todos estos aspectos se contemplan a nivel de Modelos Independientes de Computación (CIM Computation Independent Model), PIM y PSM. Muestra la arquitectura simplificada de MIDAS, donde se proponen los CIM, comunes a todo el sistema, así como los PIMs y PSMs para los aspectos de contenido, hipertexto y comportamiento. También se definen las reglas (o mappings) para transformar los modelos en el mismo nivel (PIMPIM y PSMPISM) y entre distintos niveles (PIMPSM).

En MIDAS se contempla además otros aspectos a tener en cuenta en el desarrollo de un SIW y que son ortogonales, como son la arquitectura del sistema o la seguridad.

Los autores afirman en relación a los conceptos de MIDAS, que es un conjunto de diferentes conceptos que tratan de mejorar los tiempos de desarrollo, así como también de documentación mediante la implementación de modelos, más allá de los conceptos de UML.

En la investigación concluyen Zorrilla y Vela (2007) que por lo tanto, una característica diferenciadora de MIDAS es que es una metodología ligera, que se ha definido para satisfacer tanto las necesidades de los clientes como de los desarrolladores.

Además comprueban Zorrilla y Vela (2007) habitualmente, la introducción de una nueva metodología en una empresa supone que los analistas y desarrolladores tienen que cambiar sus hábitos de trabajo. Además, las necesidades más inmediatas que tiene el cliente que encarga la aplicación Web, es que su producto software esté disponible lo antes posible. El modelo de proceso iterativo e incremental aportará las ventajas de tener productos tangibles para el cliente en versiones sucesivas, además de permitir incorporar o modificar necesidades no detectadas o no planteadas en las primeras etapas de definición del producto.

Por este motivo para Vela (2003) MIDAS se ha definido con el fin de satisfacer los siguientes objetivos:

- Proporcionar a los desarrolladores una metodología basada en modelos que guíe su trabajo siguiendo su forma habitual de trabajar.
- Soportar un desarrollo de software rápido, con el fin de asegurar a los clientes una primera versión del software en el menor tiempo posible.
- Reducir la cantidad de documentación generada durante el desarrollo del SIW.

MIDAS proporcionará al cliente los productos en un tiempo corto, permitiendo introducir en cada iteración nuevos requisitos que no se identificaron en iteraciones anteriores del desarrollo. Otra ventaja será que las pruebas se harán para cada iteración con lo que se reducirá el riesgo de que se produzcan fallos. Como ya se ha dicho, según el autor Vela (2003) MIDAS propone distintas iteraciones y al final de cada una de ellas se obtiene una nueva versión del producto.

- En una primera iteración, MIDAS/SD, es la base de la metodología, conceptúa la arquitectura y define los procesos del sistema.
- En la segunda iteración denominada MIDAS/HT, se diseña el primer ejemplar del aplicativo con páginas estáticas en HTML.
- En la tercera iteración, denominada MIDAS/DB, se implementará una nueva versión del hipertexto con páginas dinámicas en XML, recibiendo como entrada el prototipo definido en la iteración previa.

- En una iteración adicional, denominada MIDAS/FC, se desarrollan los servicios y la lógica del SIW.
- En otra iteración, MIDAS/TST, se probará el sistema.

Los autores consideran corroborar que según el texto consultado se pueden encontrar diferentes iteraciones aplicadas a casos específicos, dada la característica adaptable de la metodología.

2.17.1. MODELO INDEPENDIENTE DE LA COMPUTACIÓN (CIM)

Este primer modelo representa el nivel más alto en el modelo del negocio, y describe el proceso o la lógica de la organización, para realizarlo se usa lenguaje que no sea UML.

Este modelo describe las interacciones entre los procesos y las responsabilidades de cada trabajador sea humano o no. otra característica de este modelo es que cualquier persona pueda entender el proceso del negocio. (Galván y Yesith 2013)



Figura 2.8. Diagrama UML de proceso de un proyecto

2.17.1.1. UML

UML es un lenguaje estándar de modelado para software, para la especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante. Además es un lenguaje para visualizar, mezcla gráficos y texto, pero es algo más que una simple cantidad de símbolos. De hecho, detrás de cada símbolo en la notación UML hay una semántica bien definida, de manera que un desarrollador puede escribir un modelo en UML, y otro desarrollador, e incluso otra herramienta, puede interpretar ese modelo sin ambigüedad. Básicamente, UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. (González, et. al, 2013)

2.17.2. MODELO INDEPENDIENTE DE PLATAFORMA

El modelo independiente de plataforma (Platform Independent Model o PIM) exhibe un grado de independencia tal permite mapearlo a una o varias plataformas, Esto se logra definiendo una serie de servicios abstrayéndose de los detalles técnicos para que otros modelos especifiquen cómo será la implementación. (Microsoft, 2007a).

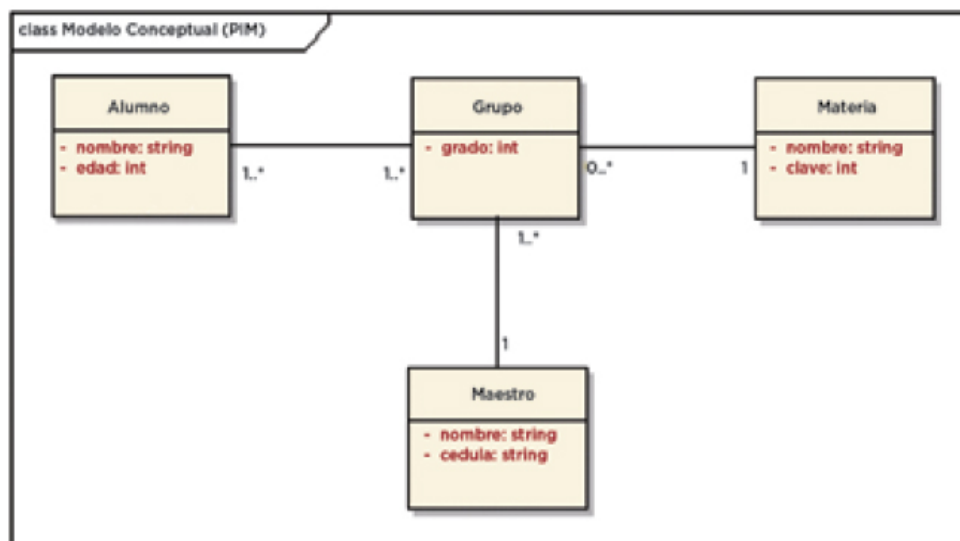


Figura 2.9. Iteración del Modelo de Independiente de plataforma.

2.17.3. MODELO ESPECÍFICO DE LA PLATAFORMA (PSM)

El modelo específico de plataforma combina la especificación de un PIM con los detalles para indicar cómo el sistema usa una plataforma en particular, tiene

un nivel de abstracción más bajo, mucho cercado a la vista de código que el PIM. Un PSM es un PIM que puede incluir más o menos detalles específicos para ser implementado en una plataforma, dependiendo de su propósito. El PSM describe la arquitectura final del sistema. Puede ser depurado sistemáticamente a niveles más bajos, hasta que pueda ser transformado a código directamente. (Armas. 2009)

2.18. INVESTIGACIÓN DOCUMENTAL

La investigación documental es la parte esencial de un proceso de investigación científica, que constituye una estrategia donde se observa y reflexiona sistemáticamente sobre realidades (teóricas o no) usando para ello diferentes tipos de documentos. Indaga, interpreta, presenta datos e informaciones sobre un tema determinado de cualquier ciencia, utilizando para ello, una metódica de análisis; teniendo como finalidad obtener resultados que pudiesen ser base para el desarrollo de la creación científica. (Anaya, 2013)

2.19. ESTANDAR IEEE 830

Este estándar no solo se encarga de poner las pautas para identificar y esquematizar los requerimientos de software como parte integral del desarrollo de software, sino también como base fundamental de este, todo esto con el fin de no caer en cambios, errores o situaciones que pongan en peligro la creación de una solución, producto o software; incurriendo en gastos o cambios producto de una mala análisis de requerimientos. (Gonzales, 2008).

CAPÍTULO III. DESARROLLO METODOLÓGICO

La tesis de grado se desarrolló en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, para la Comisión de Evaluación Interna, en la ciudad de Calceta, Cantón Bolívar, provincia de Manabí, se obtuvo como resultado un módulo web que permitió llevar el control del registro de las asistencias de los estudiantes de la universidad, se utilizó la metodología de investigación documental para recopilar la información relacionada al problema de estudio, para luego con la información recabada se aplicó la metodología Ágil MIDAS, que con la utilización de sus tres iteraciones que son el Modelo Independiente de Computación, el Modelo Independiente de la Plataforma y el Modelo Especifico de la Plataforma, se logró describir de manera abstracta cada una de las partes del módulo web de control asistencias denominado CASYS , y después se continuo con el desarrollo de la aplicación web, en el IDE desarrollo Microsoft Visual Studio 2012.

3.1. MÉTODO DE INVESTIGACIÓN DOCUMENTAL

En este método implica una recopilación de información para determinar resultados, es decir, los requerimientos necesarios para la elaboración del módulo web, se realizó una entrevista informal¹ (Anexo 1) al Director de la Comisión de Evaluación Interna, que detallo cada uno de los aspectos organizacionales de los procesos del control de asistencia, además se revisó informes, manuales de procesos, el Reglamento Integral del Desempeño del Personal Académico que analizando cada uno de estos textos, se logró entender de una manera más sistematizada como continuar con el desarrollo del mismo. Además de obtener una explicación del problema, permitió comprender la situación actual de cómo se estaban llevando estos procesos y los problemas que acarreaba al momento de presentar los informes impresos sobre las asistencias, lo que era un proceso tedioso y a la vez distorsionaba la exactitud de los datos. Como resultado de esta metodología se utilizó el formato de la norma IEEE 830 Especificación de Requerimientos de Software

¹ Vélez, A. 2014. Requerimientos del sistema (entrevista). Calceta-Manabí, EC, Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

(ERS), planteado en IEEE (2009), en que se esquematizo cada uno de los elementos que necesita el sistema.

3.2. METODOLOGÍA MIDAS

La metodología MIDAS, tiene 3 iteraciones que describieron a la aplicación web, de manera abstracta y esquemática, los diagramas se realizaron con las herramientas de UML de Visual Studio 2012.

3.2.1. PRIMERA ITERACIÓN (MODELOS INDEPENDIENTES DE COMPUTACIÓN)

En este ciclo se inició después del levantamiento de información, en que se realizaron una serie de actividades con las cuales se elaboró el modelo de dominio y el modelo de negocio representado por las entidades y objetos por los cuales estará compuesto el modulo web, las relaciones que implicará y la colaboración que existen entre ellas.

3.2.1.1. MODELO DE DOMINIO.

El modelo de dominio es diagrama de Clases, en el cual se modelo los objetos y la estructura que posee el proceso organizacional para poder realizar la aplicación, que se obtuvieron del levantamiento de información en la Especificación de Requerimientos de Software. Esto se lo realizo con la finalidad de entender con mayor detalle cual es el papel que desempeñan, tanto los objetos como las personas que trabajan dentro del sistema, describiendo cuáles son los procesos y funciones que realizan dentro del mismo.

3.2.1.2. MODELO DE NEGOCIO

El modelo de negocio son diagramas de casos de uso, que describen desde una perspectiva macro el nivel de interacción del usuario con el sistema, que se diagramó en cada caso las acciones que el usuario realizó en el aplicativo web de manera general.

3.2.2. SEGUNDA ITERACIÓN (MODELO INDEPENDIENTES DE LA PLATAFORMA)

Los siguientes modelos describen el sistema una manera formal, independiente de la implementación que esta tenga en la programación:

3.2.2.1. MODELOS CONCEPTUAL DE DATOS

El Modelo Conceptual de Datos, es una visión de los objetos planteados en el modelo de Dominio, esquematizando el flujo de información que puede existir en el sistema, tomando cada uno de los objetos como tablas de registros de información, y realizando los procedimientos almacenados para dar funcionalidad a la base de datos, mediante un diagrama de clases en la que cada una de las entidades son los objetos del sistema y dichas relaciones son las interacciones que estás tendrán.

3.2.2.2. MODELO DE SERVICIOS DE USUARIOS

El modelo de servicio es un diagrama de caso de uso, en la que se planteó cada uno de los objetos del sistema, en la que se muestra cada uno de los servicios que ofrece el aplicativo, representados de una manera general.

3.2.2.3. MODELOS DE CASOS DE USO EXTENDIDO

En esta sección se relacionaron las funciones identificando si el servicio es compuesto <<CS>>, básico funcional <<FBS>> o básico estructural <<SBS>>, es decir a los diagramas de casos de uso, se añadió estas etiquetas para identificar el tipo de servicio, correspondientes a las actividades que realiza cada uno de los procesos están directamente relacionados a los actores, además, se detalló cómo están interactuarán cada una de estas funciones entre sí.

3.2.2.4. MODELOS DE COMPOSICIÓN DE SERVICIOS

Los modelos de composición de servicios son diagrama de actividades, en los cuales se definió la secuencia de los principales servicios que se identificaron las relaciones en el modelo de caso de uso extendido, estableciendo la secuencia en que se ejecutan dichas funciones.

3.2.2.5. MODELO DE FRAGMENTOS

El objetivo del modelo de fragmentos fue para obtener una descomposición del sistema en unidades significativas de información, en que las relaciones entre estas unidades son los hipervínculos del aplicativo web, se agregan estereotipos <<SS>> fragmentos estructurales y <<FS>> que representa fragmentos funcionales, este modelo se realizó en el IDE de modelado StarUML, utilizando diagramas de estructura para mostrar la organización interna de las funciones.

3.2.2.6. MODELO DE NAVEGACIÓN

Utilizando el Modelo de Fragmentos y el Modelo de Composición de Servicios como base de conocimiento, se realizó el Modelo de Navegación, que son las relaciones de las unidades de información, describiendo menús e hipervínculos del aplicativo web, en la que el modelado se realizó en un diagrama de estructura en el IDE de StarUML.

3.2.2.7. MODELO DE SERVICIOS

El Modelo de Servicio describió de manera más interna cada uno de los servicios conceptuales con que cuenta el sistema de información Web, en el que se representan las unidades de información con sus relaciones y las funciones que intercambian entre sí, que en la herramienta de modelado StarUML se realizaron estos diagramas de comunicación, para los diferentes servicios.

3.2.3. MODELO ESPECÍFICO DE LA PLATAFORMA

Esta iteración plantea los aspectos de implementación en la tecnología a utilizar, en la que van desde los objetos-relacionales, así como las tecnologías de comunicación a implementar.

3.2.3.1. MODELO OBJETO RELACIONAL

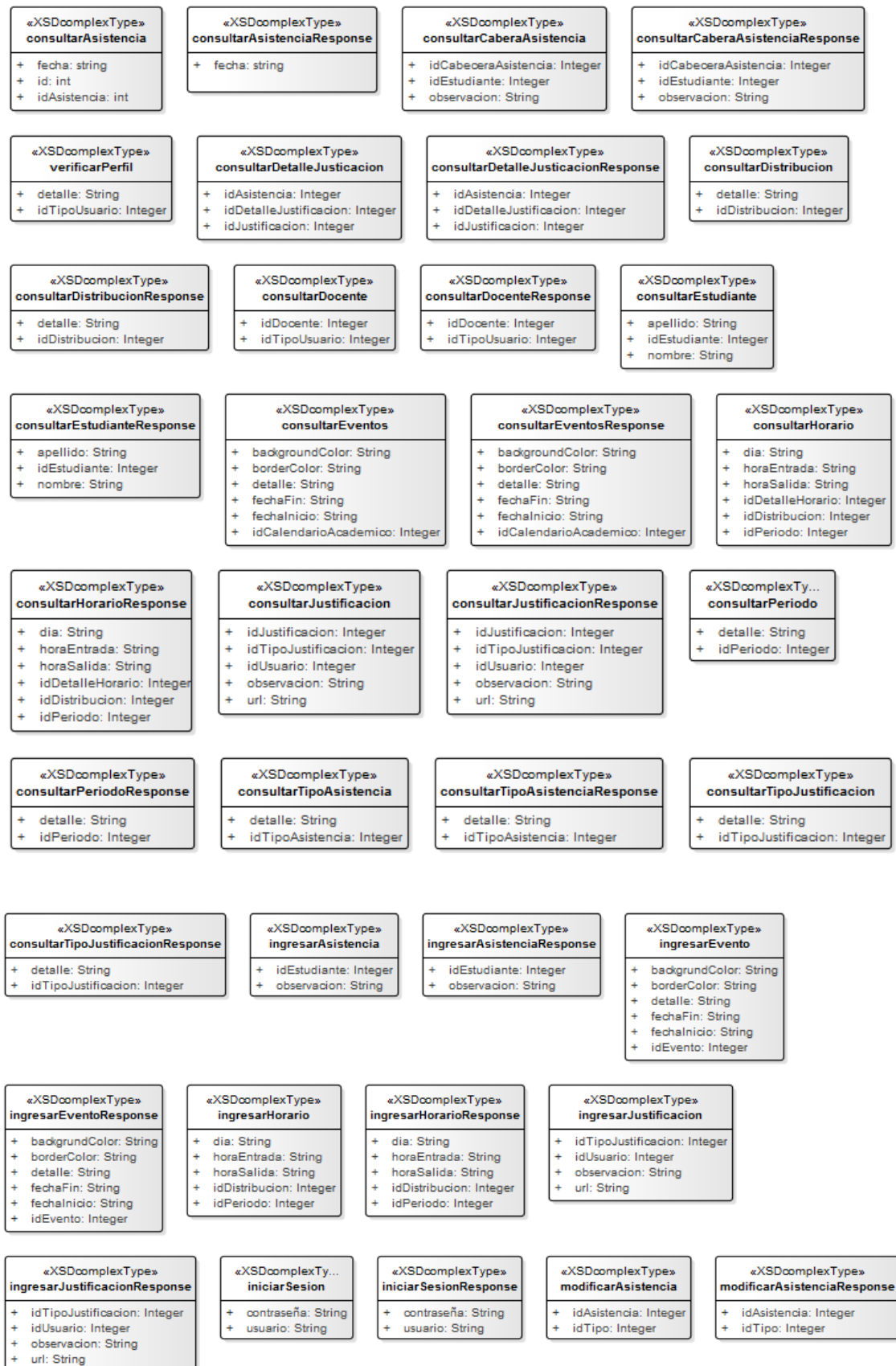
Este modelo está relacionado a nivel lógico de datos, se tomó información del Modelo Conceptual de Datos, para luego convertir las clases en unidades de información de la base de datos, tomando en cuenta las relaciones y estructuras de los tipos de datos, realizando la implementación en el gestor de base de datos Microsoft SQL Server R2.

3.2.3.2. MODELOS XML SCHEMAS.

Esta iteración contiene de los fragmentos del sistema modelados lógicamente en un documento XML, se definieron el contenido, atributos y estructura de las unidades de información descritas en el Modelo de Fragmentos, para generar este modelo se utilizó una herramienta UML Enterprise Architect de la empresa Spax Systems que cuenta con una extensión para transformar el modelo en bloques de código, en la que se definió restricciones, estereotipos, para construir un documento válido.

3.2.3.3. MODELO WSDL

Según la metodología Ágil MIDAS se deben de construir todos los servicios Web del aplicativo, indeterminadamente si estos vayan se implementen o no, por lo que el Modelo WSDL (Web Services Description Language) se utilizó para describir esta forma de comunicación en un formato XML, con una estructura en la que combinó XML Schema y SOAP, como métodos de interacción con el catálogo de funciones, el WSDL especificó la interfaz abstracta de como el usuario accede a los servicios. Para diseñar los servicios Web se utilizó Enterprise Architect un software de modelado UML de la empresa Spax Systems que cuenta con una extensión para diagramar WSDL cuya construcción en el software se realizó empezando por definir los tipos de datos, luego los mensajes que se intercambian en las funciones, se configuro las operaciones permitidas en los tipo de puertos a utilizar, se enumeró el protocolo a implementar, para concluir estableciendo los servicios del aplicación web:



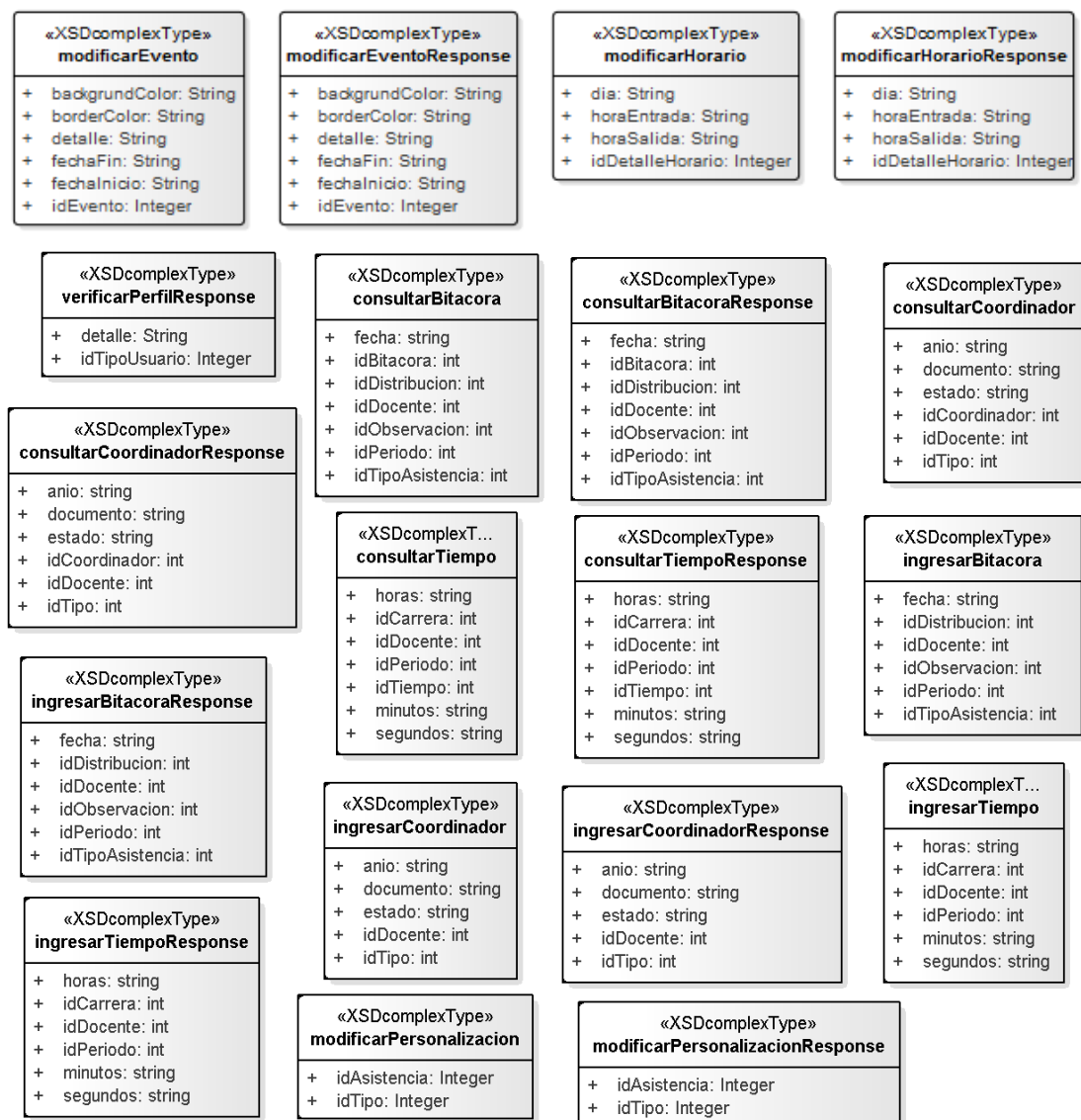
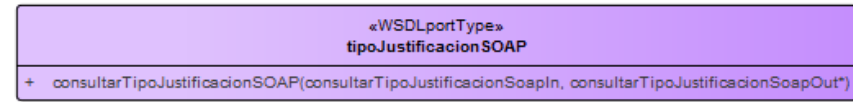
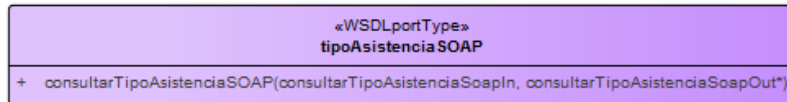
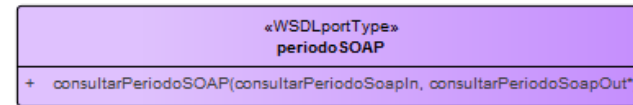
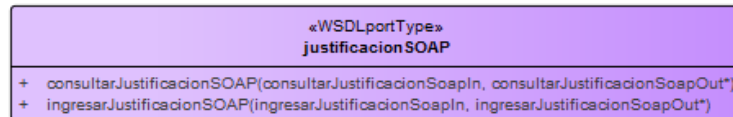
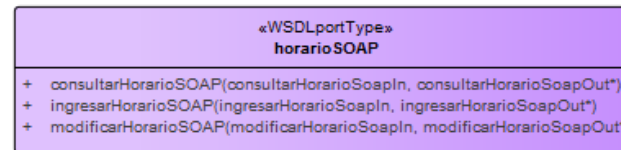
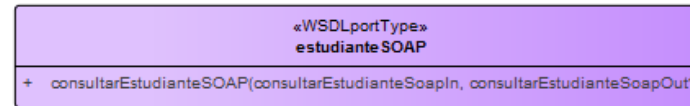
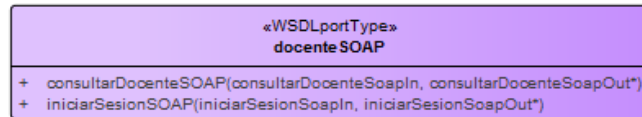
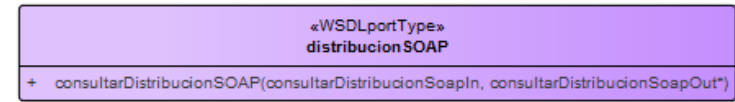
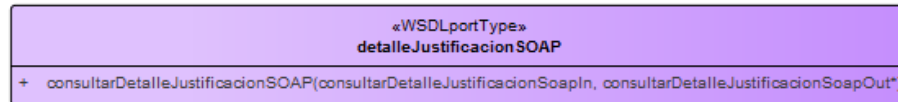
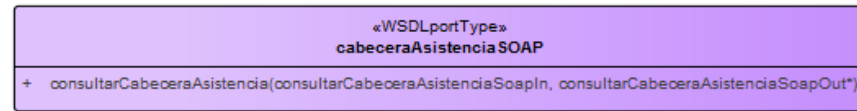
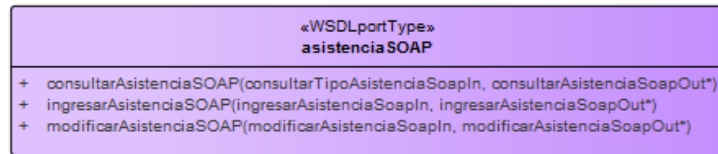


Figura 3.10. Tipos de Datos de los Servicios Web del Módulo CASYS





Figura 3.11. Mensajes que intercambian las funciones



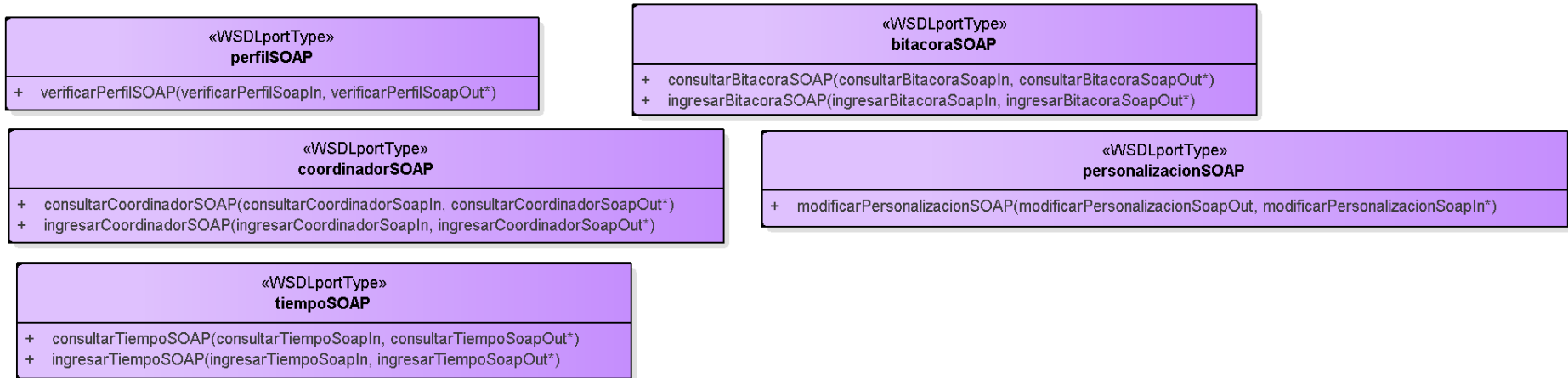
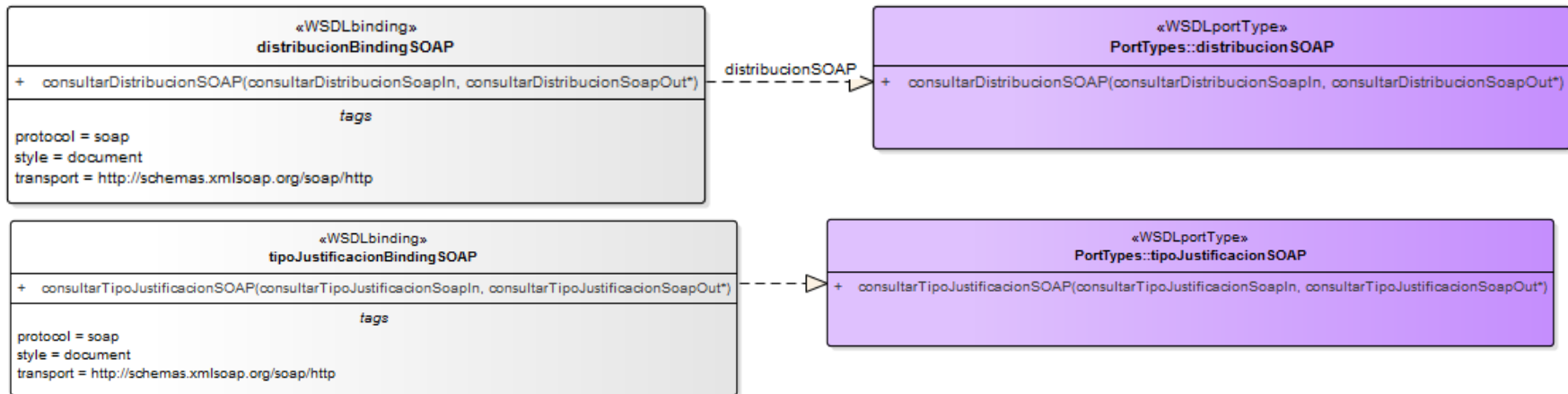
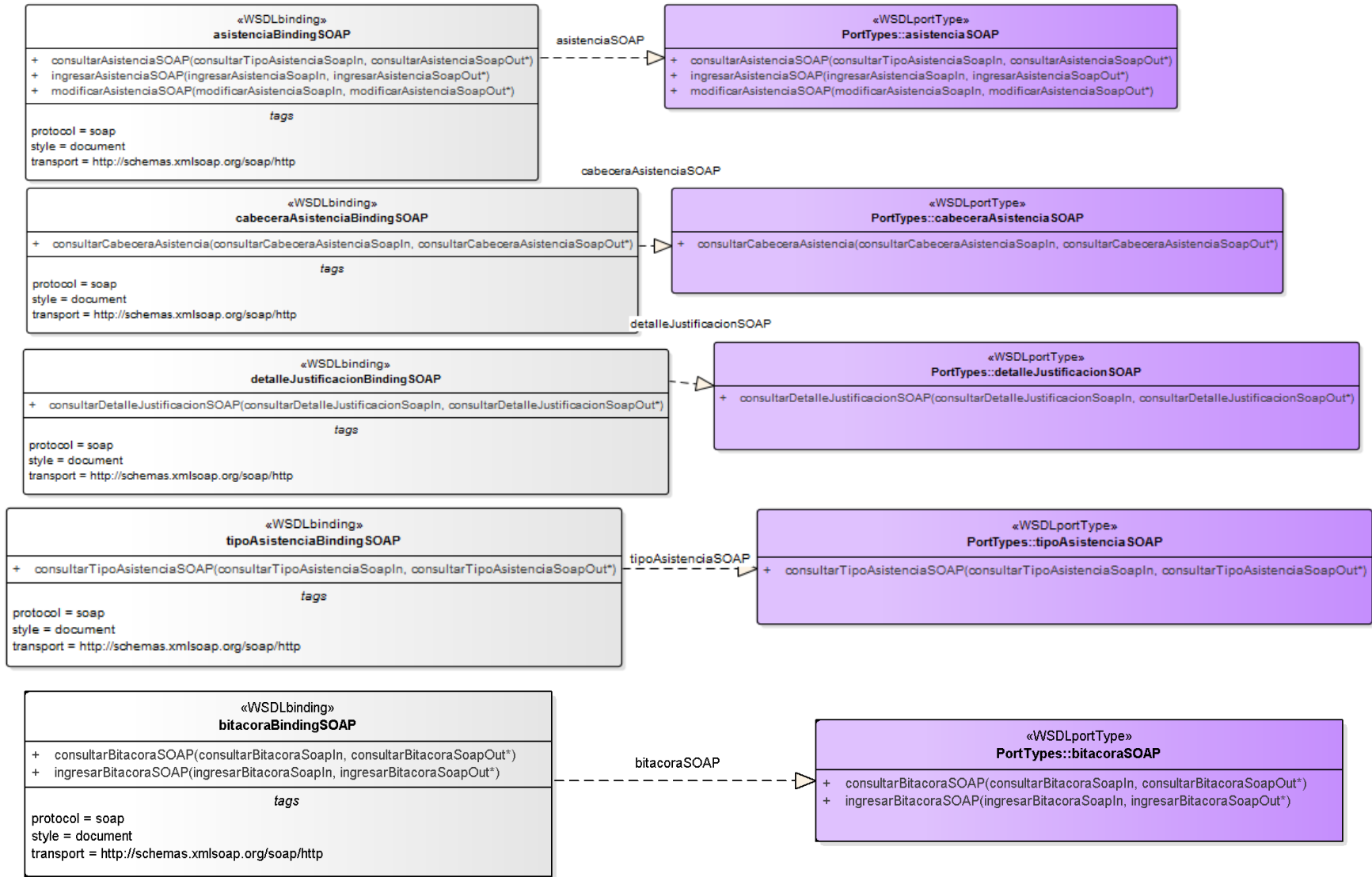
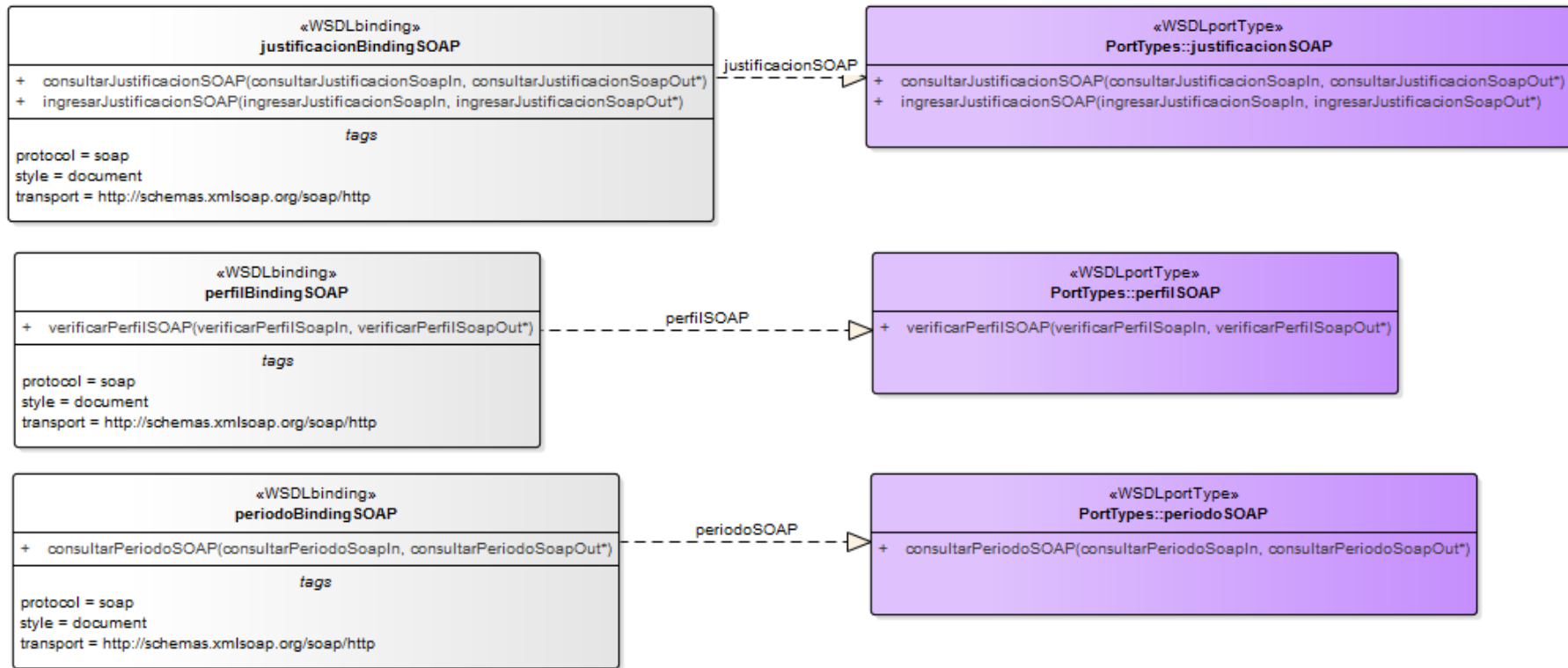


Figura 3.12. Configuración de las operaciones permitidas en los servicios SOAP.







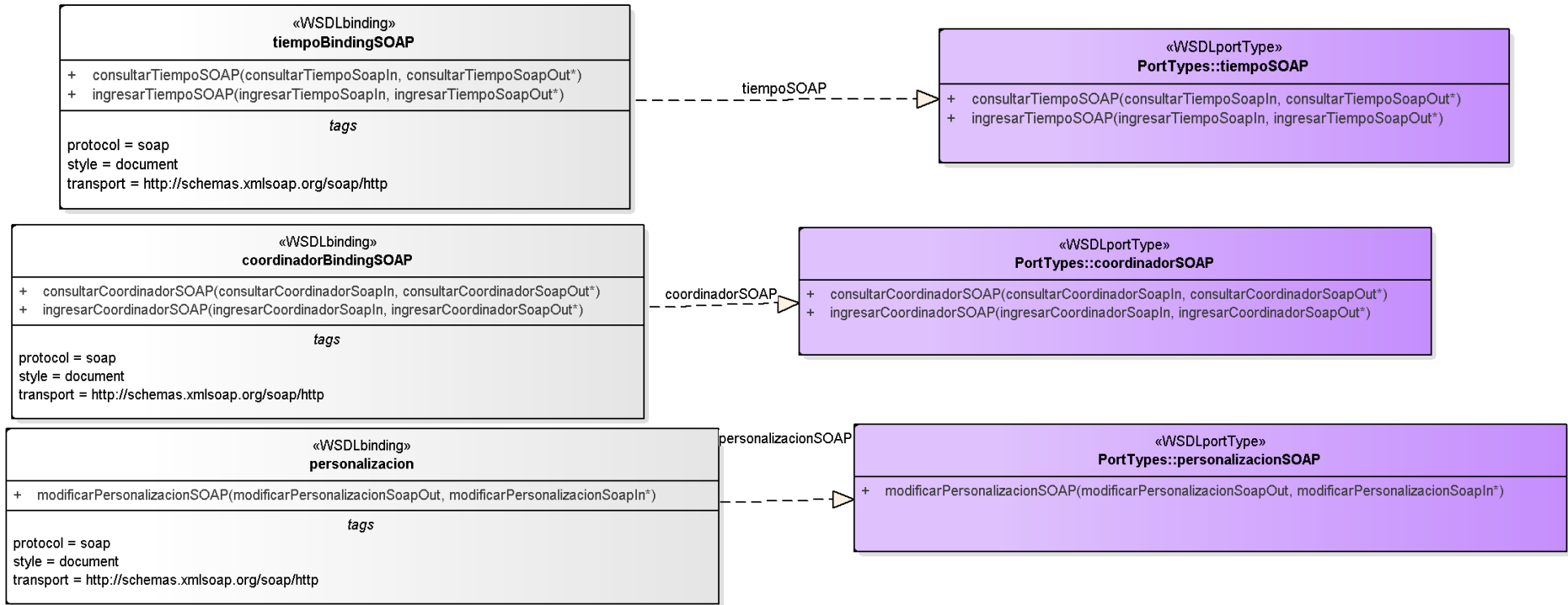
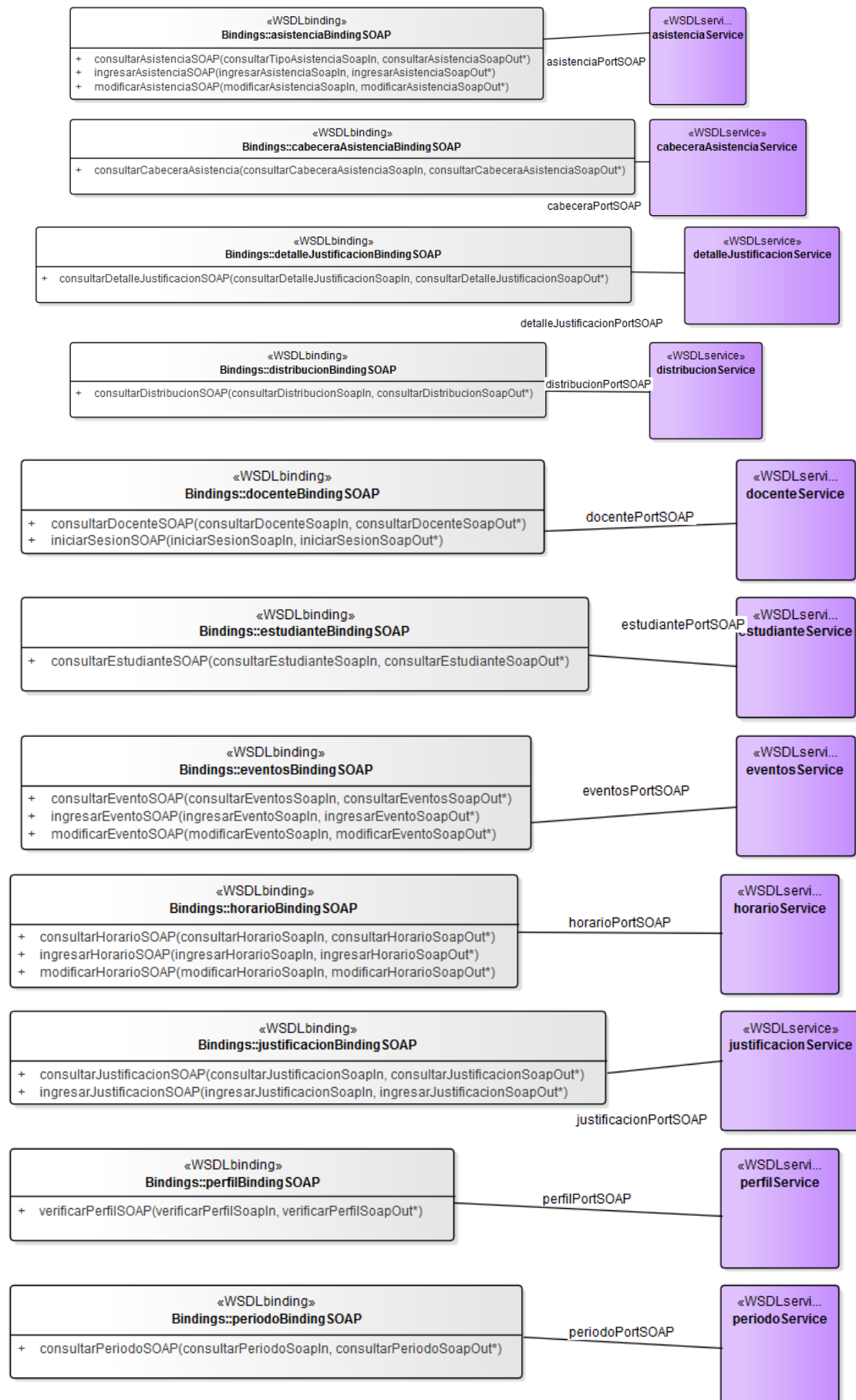


Figura 3.13. Especificación de los protocolos utilizados.



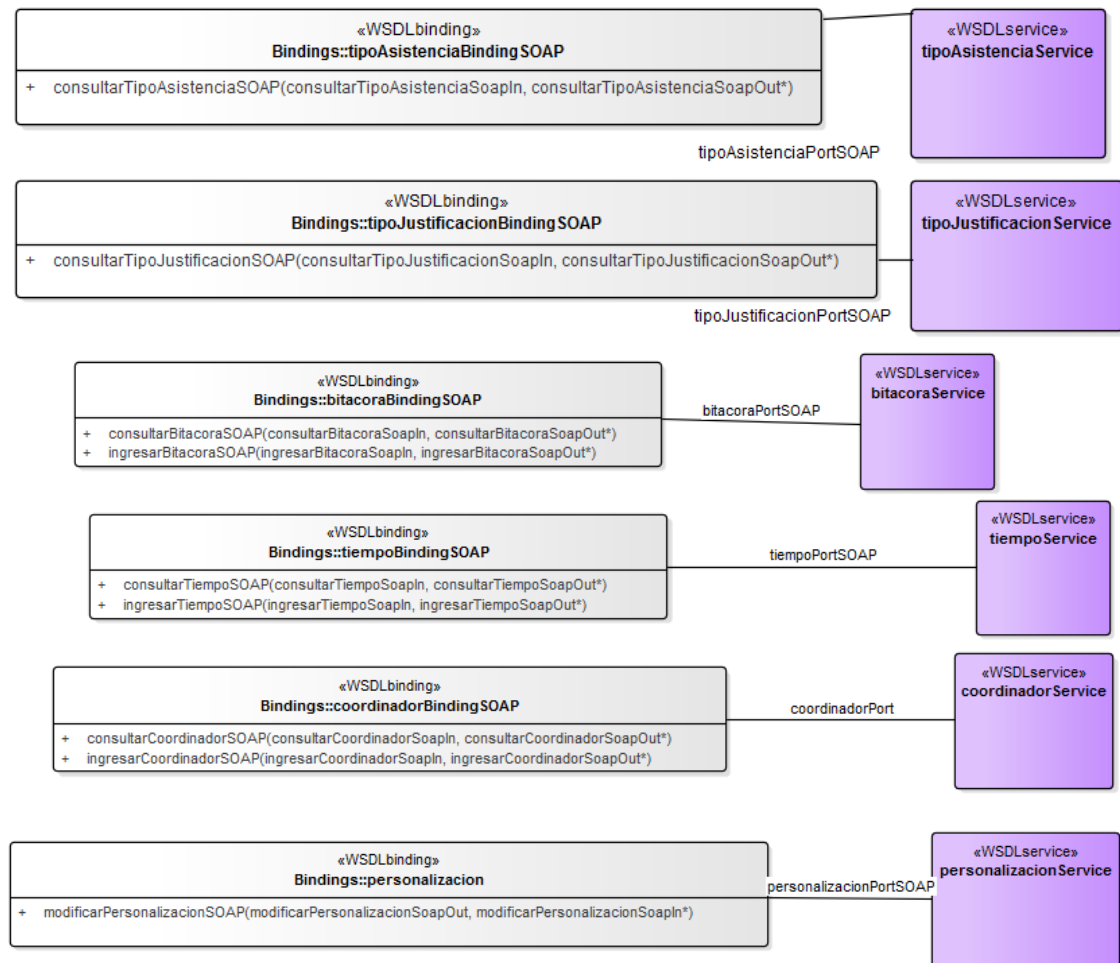


Figura 3.14. Servicios Web del Módulo de CASYS.

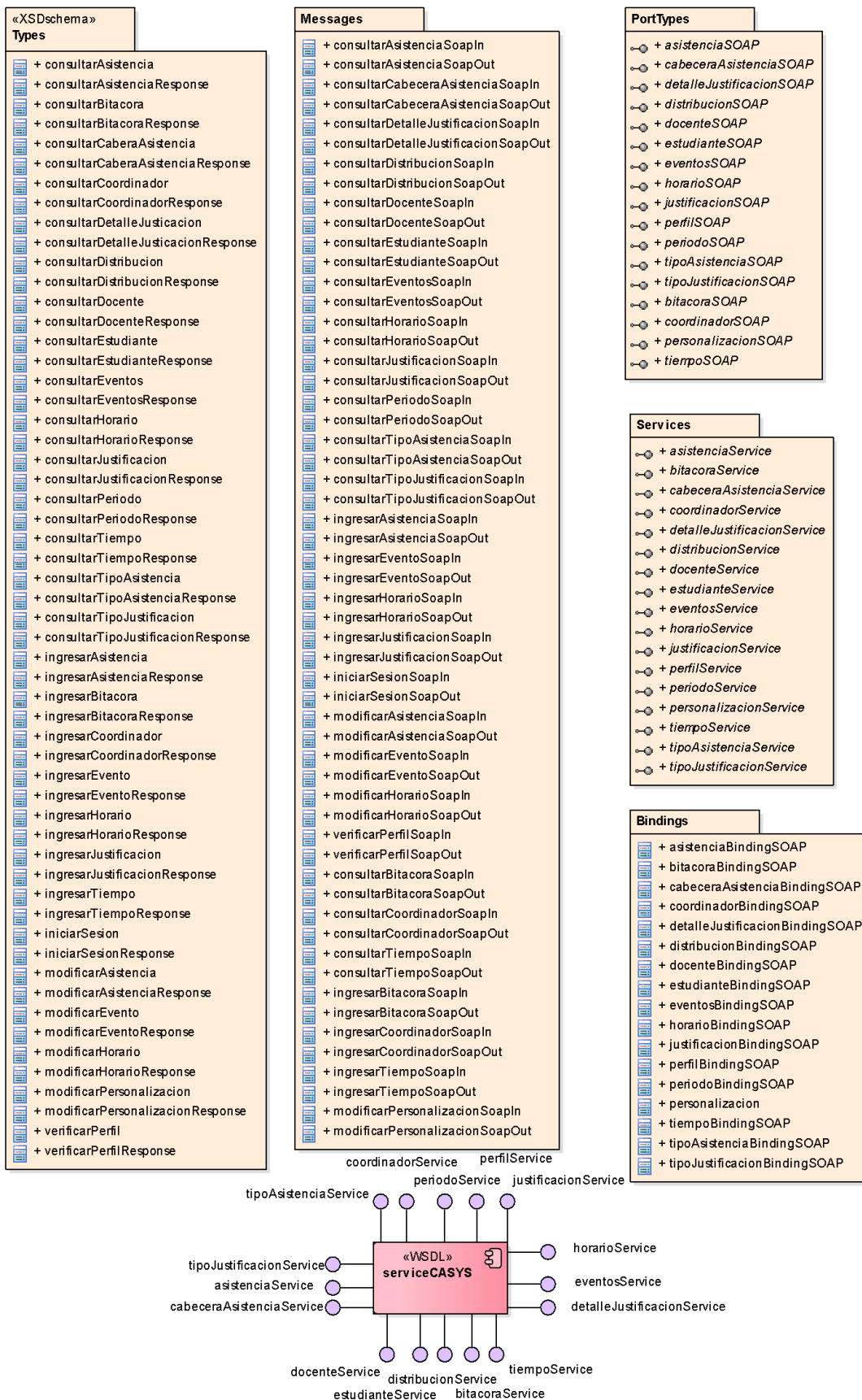


Figura 3.15. Visión de la Estructura de los Servicio Web.

3.3. DESARROLLO DEL SISTEMA

El sistema web CASYS se desarrolló con las herramientas tecnológicas, Microsoft Visual Studio 2012 como IDE para la programación del sistema y Microsoft SQL Server 2008 r2 como administrador de base de datos, en un periodo de cuatro meses, tomando como inicio el diseño obtenido en el modelo de objeto relacional, se adoptó el paradigma de programación en tres capas: Capa de datos, negocio y presentación, para separar las partes fundamentales del sistema en grupos más pequeños y poder reutilizar partes de los algoritmos diseñados.

3.3.1. PARADIGMA DE DESARROLLO POR CAPAS

Este paradigma de desarrollo implica dividir en capas las partes lógicas del sistema: es decir separar el acceso a datos o la base de datos, la lógica del negocio o la abstracción del aplicativo y el diseño o la interfaces de usuario, lo que permitió realizar en el IDE Microsoft Visual Studio 2012 una programación distribuida y fácil de reutilizar.

3.3.1.1. CAPA DE DATOS

Se diagramo las tablas con sus respectivas estructuras en el administrador de base de datos Microsoft SQL 2008, estructurando y codificando procedimientos almacenados que fueron después utilizados en la codificación, plasmando con la ayuda de herramienta de acceso a datos Entity Framework, la capa de datos que será utilizada en la capa de negocio.

3.3.1.2. CAPA DE NEGOCIO

El esquema de la capa de negocios se dividió en 2 partes: clases constructoras y clases catálogos: las clases constructoras definen los parámetros por los cuales está compuesto cada una de las tablas, por otro lado, las clases catálogos describen los métodos con los que se podrá manipular la información para cada uno de las tablas por las cuales está compuesta la base de datos. Existen una clase constructora y una clase catalogo para cada tabla en la base de datos.

3.3.1.3. CAPA DE PRESENTACION

Esta capa se relacionó con el diseño del aplicativo se tomó como base los requerimientos de la Especificación de Requerimientos de Software, en las que se relató las restricciones y sugerencias del diseño del sistema, estructurando las interfaces acorde a las necesidades del sistema, manteniendo un diseño con accesibilidad para los usuarios.

3.4. VERIFICACION DEL SISTEMA

Concluido el sistema se procedió a su respectiva publicación en el servidor de la Universidad, realizando las respectivas configuraciones de seguridad, correcciones de diseño y de funcionamiento, para luego ser validado mediante el uso del mismo con un grupo de docentes, mostrando las capacidades del sistema, y recolectando sugerencias del testing.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

En base a la metodología documental, se aplicó la norma IEEE 830, para formatear la información, por lo que se obtuvo la siguiente:

4.1. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (ERS):

4.1.1. PROPÓSITO

Este documento describió la Especificación de Requerimientos de Software (ERS), del Módulo Web de Control de Asistencias de los estudiantes de la ESPAM, en la cual se proyectó para ser utilizada como un manual durante el desarrollo y posterior implementación. También se describió cada uno de los requerimientos, que se logró obtener de la investigación realizada, las características del subsistema, lo que puede y no puede realizar, además se definieron los requerimientos tecnológicos necesarios para el buen funcionamiento del subsistema.

Esta ERS podrá ser utilizada como descripción, para obtener información sobre la administración, funcionamiento y mantenimiento, también contendrá información relevante como guía para cualquier otro desarrollador, necesite realizar mejoras o modificaciones del subsistema.

La ERS de CASYS está dirigida al personal de la Unidad de Producción de Software, cada uno de los desarrolladores y administradores del sistema.

4.1.1.1. ÁMBITO DEL SISTEMA

4.1.1.1.1. IDENTIFICACIÓN DEL SISTEMA.

CASY (Modulo de Control de Asistencia) es un módulo web que se integró al sistema general de la ESPAM, para mejorar los procesos referentes al control del registro de las asistencias de los estudiantes de la Universidad.

4.1.1.1.2. OBJETIVO DEL SISTEMA

Optimizar los procesos de presentación de reportes en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, para la mejorar el portafolio de curso de las asignaturas de la universidad.

4.1.1.1.3. MÓDULOS A CONSIDERAR.

4.1.1.1.3.1. MÓDULO DE INGRESO AL SISTEMA

Con la utilización del usuario y contraseña, podrá ingresar al subsistema, y basado en los perfiles de los usuarios se tendrá accesos a ciertos niveles del sistema. Se deberá describir a que módulos pueden ingresar los usuarios.

4.1.1.1.3.2. MÓDULO DE CALENDARIO ACADÉMICO.

Se ingresara o actualizara las fechas de los días no laborables de la universidad, así como también podrá actualizar la fecha de exámenes de la universidad.

4.1.1.1.3.3. MÓDULO DE HORARIO DE CLASES

Podrán acceder y registrar los horarios de las asignaturas del periodo actual.

4.1.1.1.3.4. MÓDULO DE ASISTENCIA

Se podrá realizar el control de la asistencia diaria de los estudiantes, en los parámetros a ingresar son contantes (número de horas asistidas o faltas).

4.1.1.1.3.5. MÓDULO DE JUSTIFICACIÓN DE ASISTENCIA.

Se puede actualizar el registro de asistencia, de los estudiantes que por motivos personales falten a clases. Se debe de subir el documento de la justificación.

4.1.1.1.3.6. MÓDULO DE GENERACIÓN DE REPORTES.

Se genera los reportes de asistencia de las diferentes asignaturas, de los profesores de la universidad. Como parámetro de ingreso será carrera, asignatura, mes y periodo.

4.1.1.1.3.7. MÓDULO DE BITÁCORA

Se puede ingresar un registro de faltas justificadas e injustificadas de los docentes para tener registro de horas efectivas trabajadas en la institución.

4.1.1.1.3.8. MÓDULO DE TIEMPO

Permite obtener el registro sobre el tiempo de uso del sistema por parte de los docentes de la institución.

4.1.1.1.3.9. MÓDULO DE COORDINADOR

Facilita el ingresos de los usuarios coordinadores, para asignar permisos de generación de reportes su año a cargo.

4.1.1.1.3.9. MÓDULO DE PERSONALIZACIÓN.

Actualiza el parámetro de número de semanas de clase de este periodo para así permitir la comparación de horas de clases.

4.1.1.2. DEFINICIONES, SIGLAS Y ABREVIATURAS.

La siguiente tabla muestra el significado de varias de las palabras más utilizadas en la Especificación de Requerimientos de Software.

Cuadro 4.1. Definiciones, siglas y abreviaturas del ERS.

Termino	Detalle
CASYS	Modulo Web desarrollado para el departamento de la ESPAM
Perfil	Descripción de las tareas a las que un usuario puede acceder, dependiendo del permiso asignado. .
IEEE	Instituto de Ingeniería Eléctrica y Electrónica, en inglés Institute of Electrical and Electronics Engineers
ESPAM	Escuela Superior Politécnica Agropecuaria de Manabí
GUI	Interfaz Gráfica de Usuario

BOOTSTRAP	Conjunto de librerías para el diseño de páginas web.
DISTRIBUCIÓN	Asignatura de los profesores.

4.1.1.3. VISIÓN GENERAL

El ERS muestra información sobre los requisitos del subsistema, de una manera general, sin describir de manera profunda el sistema, lo que permitió obtener un gran entendimiento del sistema con gran facilidad. Siguiendo el documento se detallaran los Requerimientos Específicos del sistema, de manera profunda para permitir un diseño del sistema que cumplan los requerimientos del usuario, y después poder realizar pruebas que corroboren que el sistema efectúe los requisitos planteados en este documento.

La ERS de CASY está dividida en tres temas generales:

- Introducción
- Descripción General
- Requerimientos Específicos.

4.1.2. DESCRIPCIÓN GENERAL

A continuación se detallan los factores que afectan al producto y sus requerimientos.

4.1.2.1. PERSPECTIVA DEL PRODUCTO

CASY es un módulo web que se desarrolló como requerimiento de parte de la Comisión de Evaluación Interna. El modulo se implementó en el sistema general de la ESPAM MFL, está orientado a la administración del registro de asistencia de los estudiantes de la institución.

4.1.2.1.1. INTERFACES DE SISTEMA

El sistema es dependiente y autónomo, está desarrollado utilizando una arquitectura n-Capas, manipulara los datos de manera hibrida, es decir, utilizo Servicios Web XML y una base de datos local. El diseño del aplicativo será mediante la Metodología Midas.

CASY está desarrollado sobre la plataforma Windows, el sistema manejador de base de datos que se utilizara será SQL Server 2008 R2, la GUI será desarrollada para utilizada en un browser, es decir será un aplicativo multiplataforma.

4.1.2.1.2. INTERFACES DE USUARIO

La GUI está diseñada con el framework BOOTSTRAP, lo que permite tener un diseño web adaptable, a los diferentes dispositivos existentes en el mercado. Los usuarios accederán al sistema mediante la paginas ASPX.

En el aplicativo implemento diferentes elementos para manipular la información.

- Ingreso de información mediante formularios Web.
- Métodos de validación de información en los formularios, así como también de los controles del aplicativo.
- Botones para ejecutar los diferentes procesos, como guardar, modificar, eliminar y consultar; o ejecutar funciones propias del aplicativo.
- Salidas de información mediante el uso de librerías de JavaScript.
- Generación de Reportes.

4.1.2.1.3. INTERFACES CON HARDWARE

A continuación se describirá las características lógicas de las diferentes interfaces de hardware para que se pueda ejecutar la aplicación.

TECNOLOGÍA MÍNIMA QUE DEBE DISPONER EL SERVIDOR.

Las características mínimas que debe de tener el servidor para que pueda soportar las herramientas y permita funcionar la aplicación son los siguientes:

- Procesador Pentium Dual Core 1.7. GHz.
- Memoria RAM de 1 GB.
- Disco Duro de 50 Gb-
- Tarjeta de Red 10/100 Mbps
- Monitor, mouse, teclado, CD-ROM

TECNOLOGÍA MÍNIMA QUE DEBE DISPONER LOS CLIENTES (HOST).

Las características mínimas que debe de tener los computadores de los usuarios-clientes para que pueda funcionar correctamente el modulo web:

- Procesador Pentium III 700 MHz.
- Memoria RAM de 128 Mb.
- Disco Duro de 15 Gb-
- Tarjeta de Red 10/100 Mbps
- Monitor, mouse, teclado.

4.1.2.1.4. INTERFACES CON SOFTWARE

El modulo web tendrá que disponer de alguno de los siguientes software base en el servidor:

- Sistema Operativo: Windows 2000, 2000 Server, XP, 2003 Server, Windows Server 2008, entre otros.
- Servidor Web: Internet Information Server, Versión 5.0 o Superior.
- Base de datos: SQL Server 2008 R2.

4.1.2.1.5. INTERFACES DE COMUNICACIÓN

Los equipos deben estar en un ambiente TCP/IP, con una conexión a internet.

4.1.2.1.6. RESTRICCIONES DE MEMORIA

El subsistema deberá tener como características mínimas en los equipos clientes de 128 Mb de RAM y 1 Gb mínimo en el servidor.

4.1.2.1.7. REQUERIMIENTOS DE ADECUACIÓN AL ENTORNO

El aplicativo CASY es capaz de adaptarse a los diferentes dispositivos que se utilice el usuario cliente. Los parámetros de seguridad no deberán aceptar ningún tipo de método de infiltración al sistema, como SQL Injection, Secuencia de Comandos en sitios cruzados.

Como datos iniciales se deberán tener fecha y hora del host para poder verificarla con la del servidor.

4.1.2.2. FUNCIONES DEL PRODUCTO

Las funciones que deberá contener el sistema serán los siguientes agrupados por cada módulo del sistema:

4.1.2.2.1. MÓDULO DE INGRESO AL SISTEMA

- Consultar docente en la base de datos.
- Verificar perfil de usuario.
- Validar espacios en blanco.
- Validar el campo usuario.
- Validar el campo contraseña
- Validar elementos y símbolos especiales.

4.1.2.2.2. MÓDULO DE CALENDARIO ACADÉMICO.

- Eliminar eventos
- Añadir eventos.
- Modificar eventos
- Consultar todos los eventos.
- Cerrar Sesión.

4.1.2.2.3. MÓDULO DE HORARIO DE CLASES

- Añadir horario.
- Modificar horario
- Consultar el horario de la distribución.
- Cerrar Sesión.

4.1.2.2.4. MÓDULO DE ASISTENCIA

- Consultar estudiantes.
- Consultar horario.
- Ingresar asistencia
- Guardar asistencia.
- Cerrar Sesión.

4.1.2.2.5. MÓDULO DE JUSTIFICACIÓN DE ASISTENCIA.

- Consultar estudiante.
- Consultar asistencia.
- Ingresar justificación.
- Modificar asistencia.
- Cerrar Sesión.

4.1.2.2.6. MÓDULO DE GENERACIÓN DE REPORTES.

- Consultar distribución.
- Consultar estudiantes.
- Consultar asistencias.
- Generar reporte.
- Cerrar Sesión.

4.1.2.2.7. MÓDULO DE BITÁCORA.

- Agregar bitácora
- Consultar bitácora

4.1.2.2.8. MÓDULO DE TIEMPO.

- Agregar tiempo.
- Consultar tiempo

4.1.2.2.9. MÓDULO DE COORDINADOR.

- Agregar coordinador.
- Consultar coordinador

4.1.2.2.10. MÓDULO DE PERSONALIZACIÓN.

- Agregar Personalización

4.1.2.3. CARACTERÍSTICAS DE LOS USUARIOS

El acceso al sistema está restringido por diferentes módulos mediante una verificación de su perfil en el sistema, este debe estar ingresado en la base de datos; estos niveles de acceso se agrupan en 4 niveles:

Administrador: Tiene acceso a los diferentes módulos del subsistema, se encarga de actualizar el calendario académico, además puede imprimir informes de asistencia de cualquier carrera, no pueden realizar justificaciones de estudiantes, como tampoco realizar el registro de asistencia de los estudiantes.

Directores de Carrera: Este tipo de usuario tiene permisos de justificar faltas de los estudiantes de carrera que dirigen, y a su vez pueden generar informes de asistencias del semestre de la misma carrera, solo en el caso que el mismo director dicte cátedra, puede registrar asistencia de los semestre a su cargo; puede también generar los horarios de cada una de las distribuciones de la carrera.

Coordinadores de Año: se encarga de generar informes de asistencias de los semestres que controlan, solo en el caso que el docente imparta cátedra en los mismos semestres, puede registrar asistencia de los estudiantes.

Docentes: los usuarios con esta categoría pueden registrar las asistencias de las asignaturas a su cargo, y pueden generar informes de asistencia.

Los usuarios tienen una interacción con el sistema solo en los siguientes procesos:

- Registro de asistencia de los estudiantes.
- Generación de reportes de asistencia.
- Administración de calendario académico.
- Administración de horarios de clases.
- Administración de justificación de faltas.

El sistema genera procesos automáticos que son los siguientes:

- Consulta de calendario académico.
- Consulta de horario de clases.
- Validación de campos.

- Validación de rangos en las horas de asistencia.
- Emplear formulas estadísticas para la generación de informes de asistencias.
- Verificación de los perfiles de usuarios.

4.1.2.4. RESTRICCIONES

Entre las limitaciones que se tienen en la aplicación son las siguientes:

- Utilizar como fuente de diseño del módulo web la plantilla BOOTSTRAP de la aplicación web de la universidad.
- Tomar como fuente de referencia de las formulas estadísticas el documento de Excel, de control de asistencias de los estudiantes.
- Mantener los criterios de valor para el cálculo de porcentajes de asistencias de los estudiantes.

4.1.2.5. SUPUESTOS Y DEPENDENCIAS

Las suposiciones planteadas pueden afectar al estudio de requerimientos actual:

- La generación de una base de datos en la universidad, generaría cambios de las clases así como los objetos.
- La implementación de nuevo reglamento que afecte a la asistencia de la universidad.

4.1.2.6. REQUISITOS FUTUROS.

Los requisitos planteados pueden ser posibles mejoras, que luego de estudio y análisis pueden generar cambios en el sistema:

- Mejoras en la plantilla del sistema general de la universidad.
- Implementación de nuevos mecanismos de seguridad en el ingreso del sistema.

4.1.3. REQUERIMIENTOS ESPECÍFICOS

El siguiente apartado detallara los requisitos técnicos que permitirá a los ingenieros, desarrolladores u auditores, comprobar que los procesos desarrollados con la Metodología MIDAS, estén acordes y satisfagan cada uno

de los requisitos. Será también una referencia para los usuarios finales comprendan el diseño del sistema.

4.1.3.1. INTERFACES EXTERNAS

El modulo será parte del sistema integral de la universidad, utilizara como base de diseño la plantilla BOOTSTRAP, con la misma descripción de iconos y colores.

4.1.3.2. FUNCIONES

4.1.3.2.1. MÓDULO DE INGRESO AL SISTEMA

CONSULTAR DOCENTE EN LA BASE DE DATOS.

- Introducción: Muestra los datos de los docentes en una lista.
- Nombre de la función: consultarDocente()
- Entradas: Numero de cedula.
- Proceso: Comprueba el número de identidad devolviendo los campos de código, nombre y apellido y tipo.
- Salida: Codigo del docente con sus campos.

VERIFICAR PERFIL DE USUARIO.

- Introducción: Comprueba el tipo de usuario del sistema.
- Nombre de la función: verificarPerfil()
- Entradas: Tipo Usuario.
- Proceso: Verifica el tipo de usuario consulta datos de la tabla tipoUsuario.
- Salida: Módulos a que los tendrá acceso el usuario.

VALIDAR ESPACIOS EN BLANCO.

- Introducción: Valida que los campos no tengan espacios en blancos.
- Nombre de la función: verificaCamposBlanco()
- Entradas: textBox.
- Proceso: Elimina todo tipo de espacios en blanco en los campos de usuario y contraseña.
- Salida: Campos sin espacios en blanco.

VALIDAR EL CAMPO USUARIO.

- Introducción: Valida que el campo usuario solo tenga el número máximo de caracteres, también que no contengan símbolos ni caracteres especiales.
- Nombre de la función: `verificaCampoUsuario()`
- Entradas: `textBox` (Usuario)
- Proceso: Analiza los caracteres ingresados.
- Salida: Mensaje de Error o Succes.

VALIDAR EL CAMPO CONTRASEÑA

- Introducción: Valida que el campo contraseña, no contengan símbolos ni caracteres especiales.
- Nombre de la función: `verificaCampoContrasena()`
- Entradas: `textBox` (Contraseña)
- Proceso: Analiza los caracteres ingresados.
- Salida: Mensaje de Error o Éxito.

VALIDAR ELEMENTOS Y SÍMBOLOS ESPECIALES.

- Introducción: Valida que el campo contraseña, no contengan símbolos ni caracteres especiales.
- Nombre de la función: `verificaCaracteresEspeciales()`
- Entradas: `textBox` (Contraseña)
- Proceso: Analiza los caracteres ingresados.
- Salida: Mensaje de Error o Éxito.

4.1.3.2.2. MÓDULO DE CALENDARIO ACADÉMICO.**ELIMINAR EVENTOS**

- Introducción: Elimina el registro del evento en la base de datos.
- Nombre de la función: `eliminaEvento()`
- Entradas: Código de Evento.
- Proceso: Luego de confirmar de eliminar, se elimina el registro de la base de datos.
- Salida: Mensaje de Error o exito.

AÑADIR EVENTOS.

- Introducción: Añade el registro del evento en la base de datos.
- Nombre de la función: agregarEvento()
- Entradas: fecha, detalle y color.
- Proceso: Luego de arrastrar el evento al calendario, se agregan los datos a la base de datos.
- Salida: Mensaje de Error o éxito.

MODIFICAR EVENTOS

- Introducción: Modifica el registro del evento en la base de datos.
- Nombre de la función: modificaEvento()
- Entradas: código de evento, fecha, detalle y color.
- Proceso: Luego de seleccionar el evento al calendario, se reescriben los datos en la base de datos.
- Salida: Mensaje de Error o éxito.

CONSULTAR TODOS LOS EVENTOS.

- Introducción: Muestra los eventos de la tabla Calendario Academico de la base de datos.
- Nombre de la función: consultarEvento()
- Entradas: No necesita ninguna entrada.
- Proceso: Después de llamar a la función, se muestran los registros en una lista.
- Salida: Registros de la base de datos.

CERRAR SESIÓN.

- Introducción: Destruye la sesión activa de los usuarios.
- Nombre de la función: cerrarSesion()
- Entradas: No necesita ninguna entrada.
- Proceso: Después de verificar la sesión activa y después la destruye.
- Salida: Mensaje de Éxito o Error.

4.1.3.2.3. MÓDULO DE HORARIO DE CLASES

AÑADIR HORARIO.

- Introducción: Añade el registro del horario de clase en la base de datos.
- Nombre de la función: agregarHorarioClase()
- Entradas: código de distribución horaInicio, horaFin, Dia, detalle y color.
- Proceso: Valida los datos, inserta los datos en la base de datos.
- Salida: Mensaje de Error o éxito.

MODIFICAR HORARIO

- Introducción: Modifica el registro del horario de clase en la base de datos.
- Nombre de la función: modificaHorarioClase ()
- Entradas: código de evento, fecha, detalle y color.
- Proceso: Luego de seleccionar el evento al horario, se reescriben los datos en la base de datos.
- Salida: Mensaje de Error o éxito.

CONSULTAR EL HORARIO DE LA DISTRIBUCIÓN.

- Introducción: Muestra la distribución del docente.
- Nombre de la función: consultarDistribucion()
- Entradas: Código Docente.
- Proceso: Selecciona todos los registros de los campos de la tabla distribución.
- Salida: Registros de la base de datos.

CONSULTAR PERIODO

- Introducción: Muestra el periodo actual del sistema.
- Nombre de la función: consultarPeriodo()
- Entradas: Código de Distribución.
- Proceso: Selecciona todos los registros de los campos de la tabla Periodo.
- Salida: Periodo lectivo actual.

CERRAR SESIÓN.

- Introducción: Destruye la sesión activa de los usuarios.
- Nombre de la función: cerrarSesion()
- Entradas: No necesita ninguna entrada.
- Proceso: Después de verificar la sesión activa y después la destruye.

- Salida: Mensaje de Éxito o Error.

4.1.3.2.4. MÓDULO DE ASISTENCIA

CONSULTAR ESTUDIANTES.

- Introducción: Muestra el registro de estudiantes de una distribución.
- Nombre de la función: consultarEstudiantes()
- Entradas: Código Distribución.
- Proceso: Filtra los registros de estudiantes mediante el código de distribución. Ordena alfabéticamente.
- Salida: Registros de la base de datos.

CONSULTAR HORARIO.

- Introducción: Muestra el registro del horario de clase.
- Nombre de la función: consultarHorario()
- Entradas: Código Distribución, Código periodo.
- Proceso: Filtra los registros del horario de clase mediante el código de distribución y el periodo. Ordena alfabéticamente.
- Salida: Registros de la base de datos.

INGRESAR ASISTENCIA.

- Introducción: Realiza el ingreso de la asistencia de los estudiantes.
- Nombre de la función: ingresoAsistencia()
- Entradas: Código Distribución, código estudiante, valor de asistencia.
- Proceso: Se validan los datos ingresados como el valor de asistencia con el número de hora clase; se inserta los datos en la base de datos.
- Salida: Mensaje de Error o éxito.

CERRAR SESIÓN.

- Introducción: Destruye la sesión activa de los usuarios.
- Nombre de la función: cerrarSesion()
- Entradas: No necesita ninguna entrada.
- Proceso: Después de verificar la sesión activa y después la destruye.
- Salida: Mensaje de Éxito o Error.

4.1.3.2.5. MÓDULO DE JUSTIFICACIÓN DE ASISTENCIA.

CONSULTAR ESTUDIANTE.

- Introducción: Muestra el registro de estudiantes de un código único.
- Nombre de la función: consultarEstudiantesCedula()
- Entradas: Numero de cedula.
- Proceso: Filtra los registros de estudiantes mediante el número de cedula. Ordena alfabéticamente.
- Salida: Código de Estudiante.

CONSULTAR ASISTENCIA.

- Introducción: Muestra el registro de asistencia de un estudiante.
- Nombre de la función: consultarAsistencia()
- Entradas: Código de estudiante.
- Proceso: Filtra el registro de asistencia del estudiante mediante el código del estudiante. Ordena alfabéticamente.
- Salida: Registro de asistencia.

INGRESAR JUSTIFICACIÓN.

- Introducción: Ingresa el registro de la justificación de asistencia.
- Nombre de la función: ingresarJustificacion()
- Entradas: Fecha de justificación, código de estudiante, observación.
- Proceso: Valida los datos, inserta los datos en la base de datos.
- Salida: Mensaje de Error o éxito.

MODIFICAR ASISTENCIA.

- Introducción: Modifica el registro de la asistencia, cambiando la falta de la justificación.
- Nombre de la función: modificaAsistencia ()
- Entradas: Código asistencia, numero de hora.
- Proceso: Luego de seleccionar el registro de la asistencia, se reescriben los datos en la base de datos.
- Salida: Mensaje de Error o éxito.

CERRAR SESIÓN.

- Introducción: Destruye la sesión activa de los usuarios.
- Nombre de la función: cerrarSesion()
- Entradas: No necesita ninguna entrada.
- Proceso: Después de verificar la sesión activa y después la destruye.
- Salida: Mensaje de Éxito o Error.

4.1.3.2.6. MÓDULO DE GENERACIÓN DE REPORTES.**CONSULTAR DISTRIBUCIÓN.**

- Introducción: Muestra la distribución del docente.
- Nombre de la función: consultarDistribucion()
- Entradas: Código Docente.
- Proceso: Selecciona todos los registros de los campos de la tabla distribución.
- Salida: Registros de la base de datos

CONSULTAR ESTUDIANTES.

- Introducción: Muestra el registro de estudiantes de una distribución.
- Nombre de la función: consultarEstudiantes()
- Entradas: Código Distribucion.
- Proceso: Filtra los registros de estudiantes mediante el código de distribución. Ordena alfabéticamente.
- Salida: Registros de la base de datos.

CONSULTAR ASISTENCIAS.

- Introducción: Muestra el registro de asistencia de un estudiante.
- Nombre de la función: consultarAsistencia()
- Entradas: Código de estudiante.
- Proceso: Filtra el registro de asistencia del estudiante mediante el código del estudiante. Ordena alfabéticamente.
- Salida: Registro de asistencia.

GENERAR REPORTE.

- Introducción: Genera el reporte de la información de las asistencias de los estudiantes.
- Nombre de la función: generarReporte()
- Entradas: Código de Distribución, mes y año.
- Proceso: Filtra el registro de asistencia mediante el código de la distribución, se imprime los datos en pantalla. Ordena alfabéticamente.
- Salida: Reporte.

CERRAR SESIÓN.

- Introducción: Destruye la sesión activa de los usuarios.
- Nombre de la función: cerrarSesion()
- Entradas: No necesita ninguna entrada.
- Proceso: Después de verificar la sesión activa y después la destruye.
- Salida: Mensaje de Éxito o Error.

4.1.3.2.7. MÓDULO DE BITÁCORA.**CONSULTAR BITÁCORA**

- Introducción: Muestra el registro de faltas de los docentes.
- Nombre de la función: consultarBitacora()
- Entradas: Código de la carrera.
- Proceso: Filtra el registro de faltas de los docentes mediante el código de la carrera.
- Salida: Registro de faltas.

AGREGAR BITÁCORA

- Introducción: Agrega un registro de faltas de los docentes.
- Nombre de la función: agregarBitacora()
- Entradas: Código de Docente, .fecha, código de distribución, mensaje de observación.
- Proceso: valida los datos, y luego crea un registro en la base de datos.
- Salida: Mensaje de éxito o error.

4.1.3.2.8. MÓDULO DE TIEMPO.

CONSULTAR TIEMPO

- Introducción: Muestra el registro del uso del sistema por partes de los docentes.
- Nombre de la función: consultarTiempo()
- Entradas: Código de la carrera.
- Proceso: Filtra el registro del tiempo de uso de los docentes mediante el código de la carrera.
- Salida: Registro del tiempo.

AGREGAR TIEMPO

- Introducción: Agrega un registro del tiempo de uso del sistema.
- Nombre de la función: agregarTiempo()
- Entradas: Código de Docente, .fecha, tiempo, código de periodo.
- Proceso: valida los datos, y luego crea un registro en la base de datos.
- Salida: Mensaje de éxito o error.

4.1.3.2.9. MÓDULO DE COORDINADOR.**CONSULTAR COORDINADOR**

- Introducción: Muestra el registro de usuarios que tienen asignados permisos de coordinadores.
- Nombre de la función: consultarCoordinador()
- Entradas: Código de Periodo.
- Proceso: Filtra el registro de usuarios mediante el código del periodo.
- Salida: Registro del tiempo.

AGREGAR COORDINADOR

- Introducción: Agrega un registro del usuario coordinador en la base de datos.
- Nombre de la función: agregarCoordinador()
- Entradas: Código de Docente, .fecha, tiempo, código de periodo
- Proceso: valida los datos, y luego crea un registro en la base de datos.
- Salida: Mensaje de éxito o error.

4.1.3.2.10. MÓDULO DE PERSONALIZACIÓN.

AGREGAR PERSONALIZACIÓN

- Introducción: Agrega un registro del números de semanas, periodo de clases.
- Nombre de la función: agregarPersonalizacion()
- Entradas: Código de periodo, .fecha, número de semanas.
- Proceso: valida los datos, y luego crea un registro en la base de datos.
- Salida: Mensaje de éxito o error.

4.1.3.3. REQUERIMIENTOS DEL BANCO DE DATOS LÓGICOS.

Los objetos con los que el sistema interactuara son los siguientes:

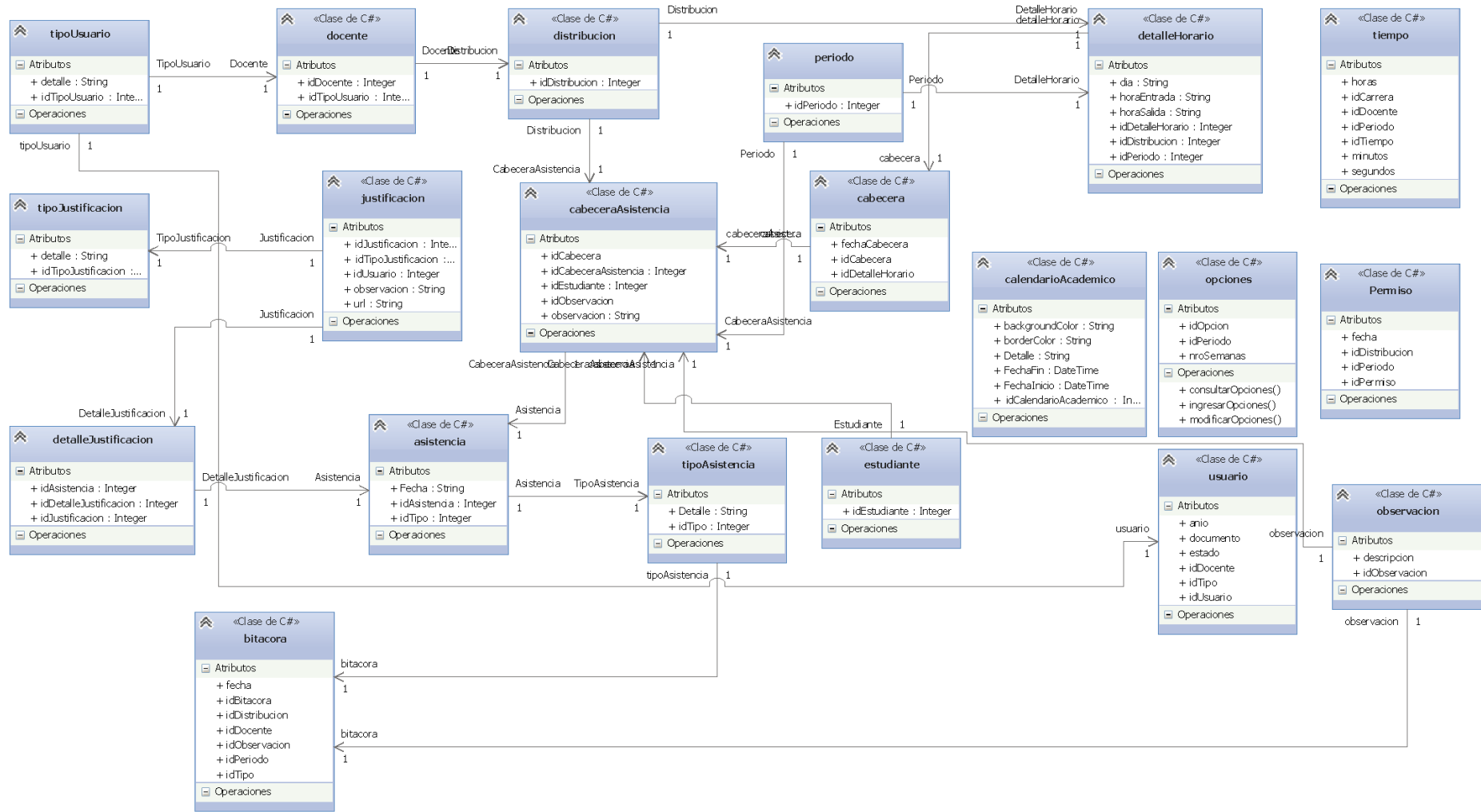


Figura 4.16. Datos del banco Lógico.

4.1.3.4. RESTRICCIONES DE DISEÑO.

- El sistema aplicara el estilo de escritura CamelCase, de tipo lowerCamelCase en al que los nombres de cada de parte del sistema empezara la primera letra con minúscula.
 - Ejemplo: generarReporte().
- Los nombres de las variables del sistema tienen que tener relación con el contenido y utilidad de las mismas.
 - Ejemplo: numeroDeHoras.
- Los controles del sistema utilizaran 3 letras minúsculas que indican el tipo de control, seguidas de otras letras (la primera mayúscula), que harán referencia al uso que se va a dar a dicho control.
 - Ejemplo: btnAceptar (Botón).

4.2. METODOLOGIA MIDAS

Con la utilización de la metodología MIDAS se obtuvo una serie de resultados, que ayudaron a describir cada uno de los componentes de las iteraciones que plantea en los siguientes modelos:

4.2.1. MODELO INDEPENDIENTE DE COMPUTACIÓN

MIDAS en su primera iteración se obtuvo descripciones generales de los objetos con sus funciones que intervienen en el sistema, además se obtuvo de manera general de principales funciones del sistema.

4.2.1.1. MODELO DE DOMINIO

Este modelo describió la estructura de los objetos que interactúan con las funciones de cada de uno de las entidades con las que cuenta el aplicativo web.

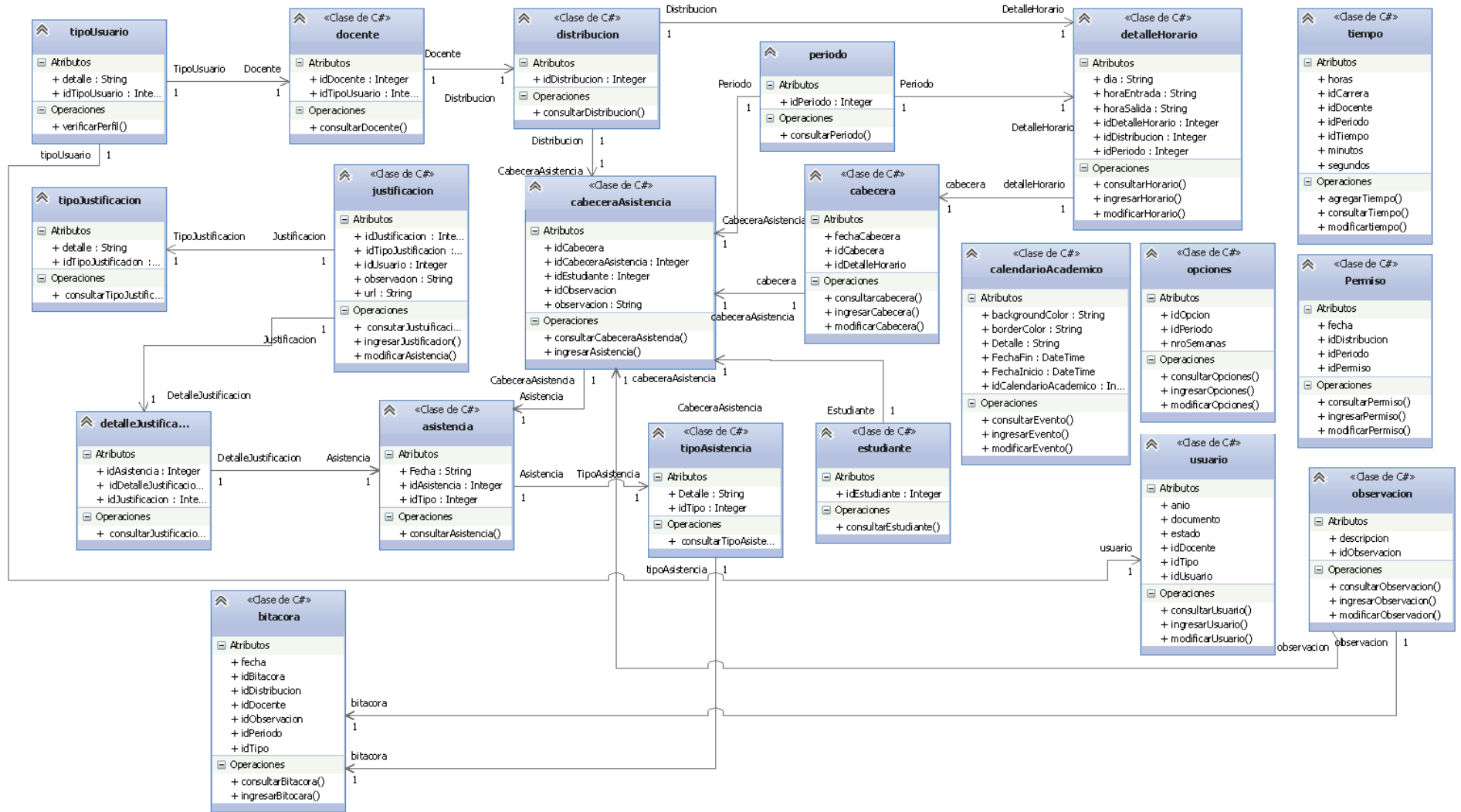


Figura 4.17. Modelo de Dominio

4.2.1.2. MODELO DE NEGOCIO

El modelo de negocio plasma las funciones principales del módulo web, desde una perspectiva general y genérica.

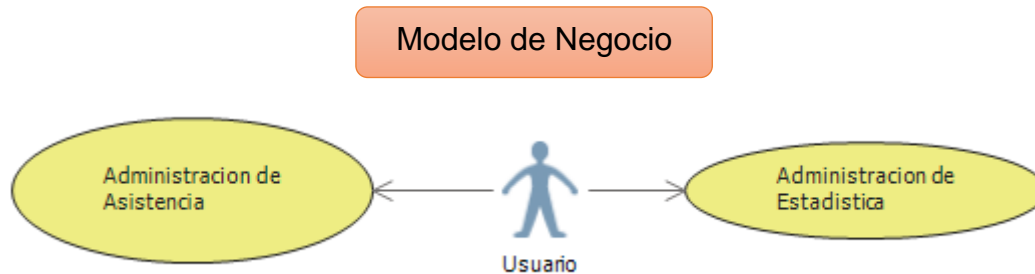


Figura 4.18. Modelo de Negocio

4.2.2. MODELO INDEPENDIENTES DE LA PLATAFORMA.

Los modelos de esta iteración muestran, especificaciones que se utilizarán en el desarrollo de la programación del aplicativo web.

4.2.2.1. MODELOS CONCEPTUAL DE DATOS

Este modelo presenta el flujo de información que tendrá a través de las entidades del sistema.

4.2.2.2. MODELO DE SERVICIOS DE USUARIOS

Este modelo describió los segmentos del módulo web, las funciones que realiza cada uno de ellos, lo que brinda una especificación clara de las acciones de los actores en el sistema.

4.2.2.2.1. MODELO DE SERVICIOS DE USUARIO - ADMINISTRADOR

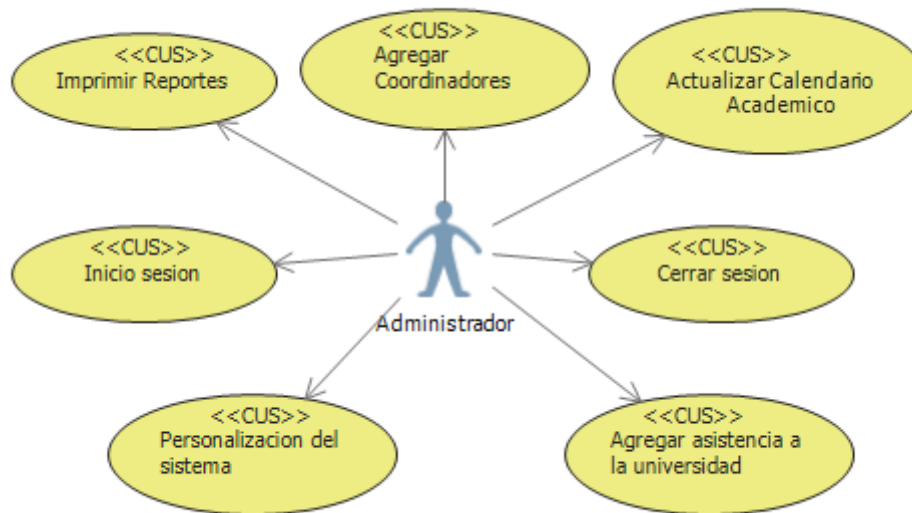


Figura 4.20. Modelo de Servicios del Administrador

4.2.2.2.2. MODELO DE SERVICIOS DE USUARIO – COORDINADOR DE AÑO

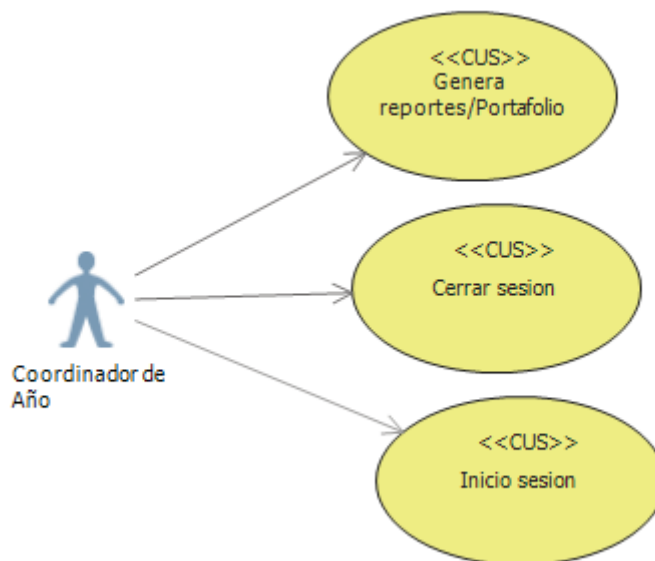


Figura 4.21. Modelo de Servicios del Coordinador de año

4.2.2.2.3. MODELO DE SERVICIOS DE USUARIO – DIRECTOR DE CARRERA.

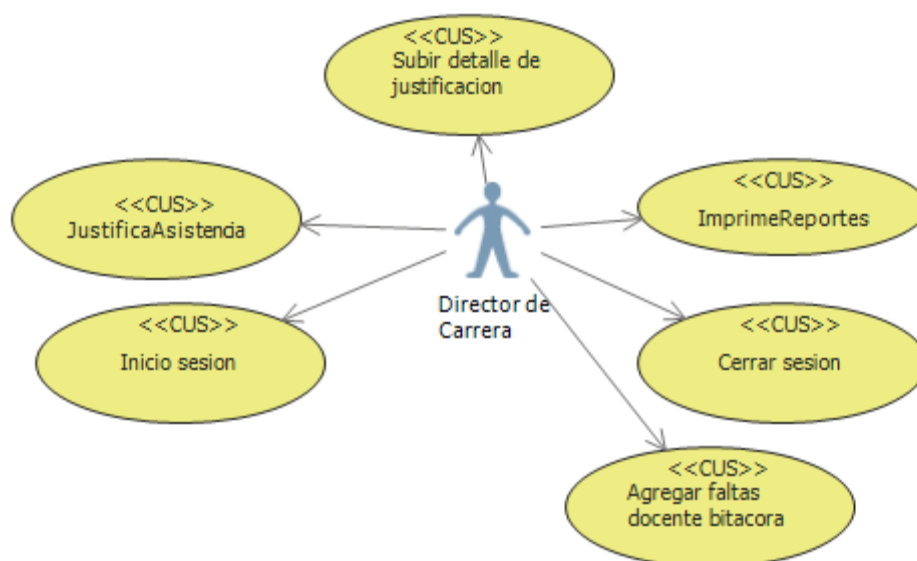


Figura 4.22. Modelo de Servicios del Director de Carrera

4.2.2.2.4. MODELO DE SERVICIOS DE USUARIO – ESTUDIANTE.

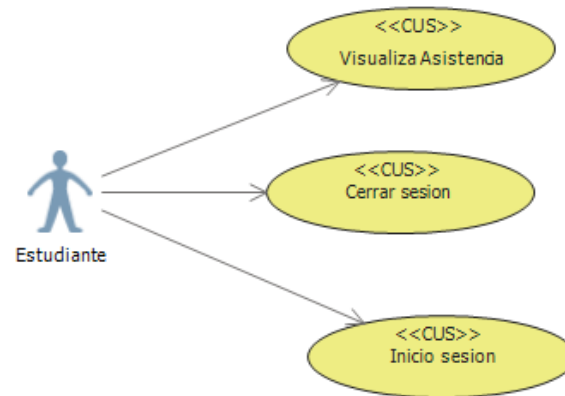


Figura 4.23. Modelo de Servicios del Estudiante

4.2.2.2.5. MODELO DE SERVICIOS DE USUARIO – DOCENTE.

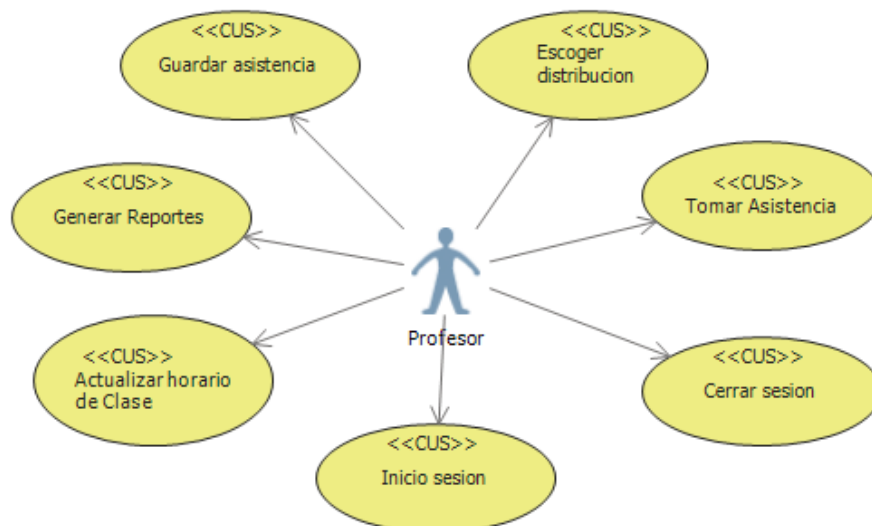


Figura 4.24. Modelo de Servicios del Docente

4.2.2.2.6. MODELO DE SERVICIOS DE USUARIO – ASISTENCIA

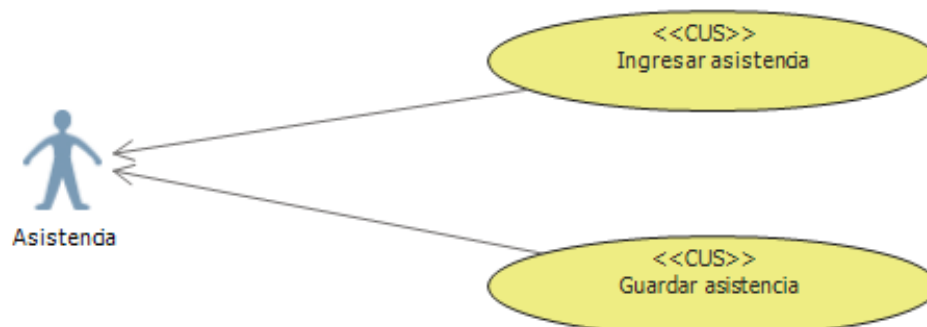


Figura 4.25. Modelo de Servicios Asistencia

4.2.2.2.7. MODELO DE SERVICIOS DE USUARIO – CALENDARIO ACADÉMICO

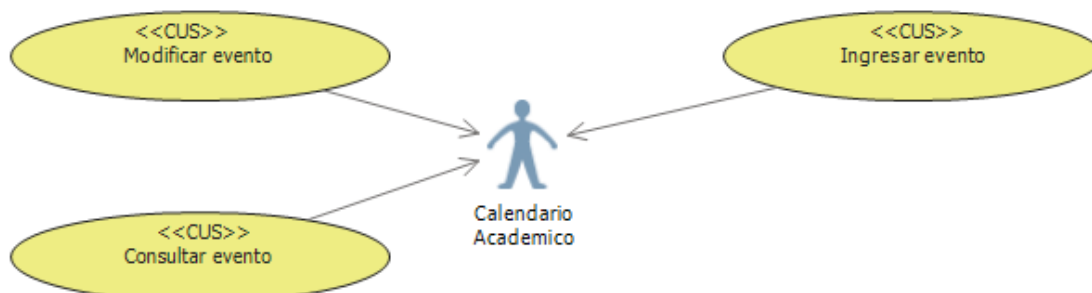


Figura 4.26. Modelo de Servicios Calendario Académico

4.2.2.2.8. MODELO DE SERVICIOS DE USUARIO – HORARIO DE CLASES

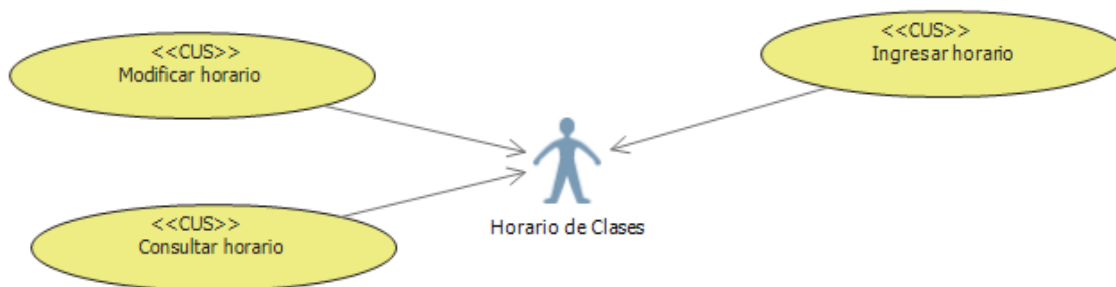


Figura 4.27. Modelo de Servicios del Horario de Clases

4.2.2.2.9. MODELO DE SERVICIOS DE USUARIO – JUSTIFICACIÓN

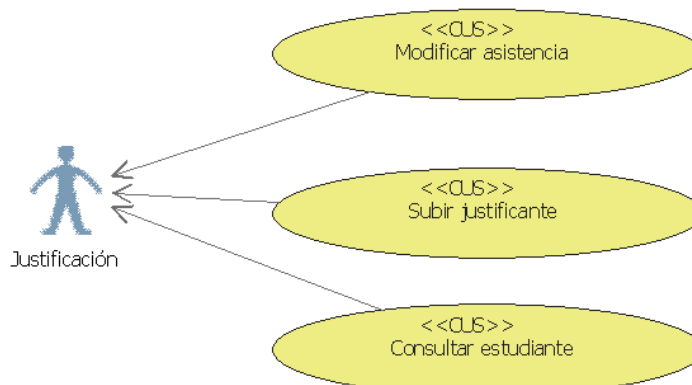


Figura 4.28. Modelo de Servicios del Horario de Clases

4.2.2.2.10. MODELO DE SERVICIOS DE USUARIO – REPORTE

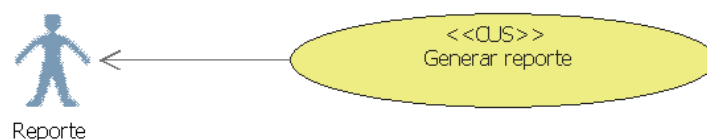


Figura 4.29. Modelo de Servicios del Horario de Clases

4.2.2.3. MODELOS DE CASOS DE USO EXTENDIDO

Con la ayuda del Modelo de Servicios de Usuario en la que muestran las funciones individuales por actores en el sistema, se pudo relacionar las funciones con sus respectivas dependencias, mostrando cómo interactúan los actores entre sí.

4.2.2.3.1. MODELO DE CASOS DE USOS EXTENDIDO- ADMINISTRADOR.

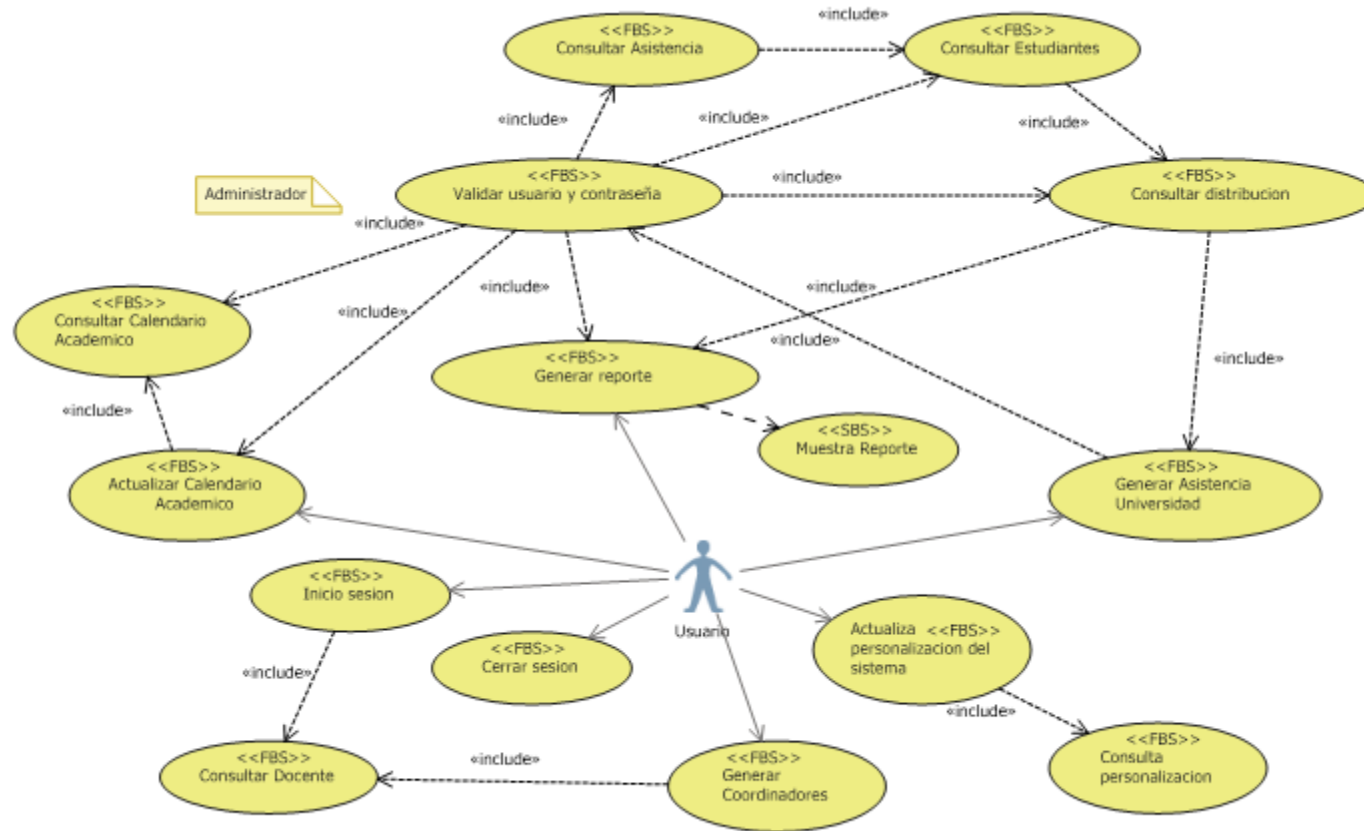


Figura 4.30. Modelo de Caso de Uso Extendido - Administrador

4.2.2.3.2. MODELO DE CASOS DE USOS EXTENDIDO- COORDINADOR DE AÑO.

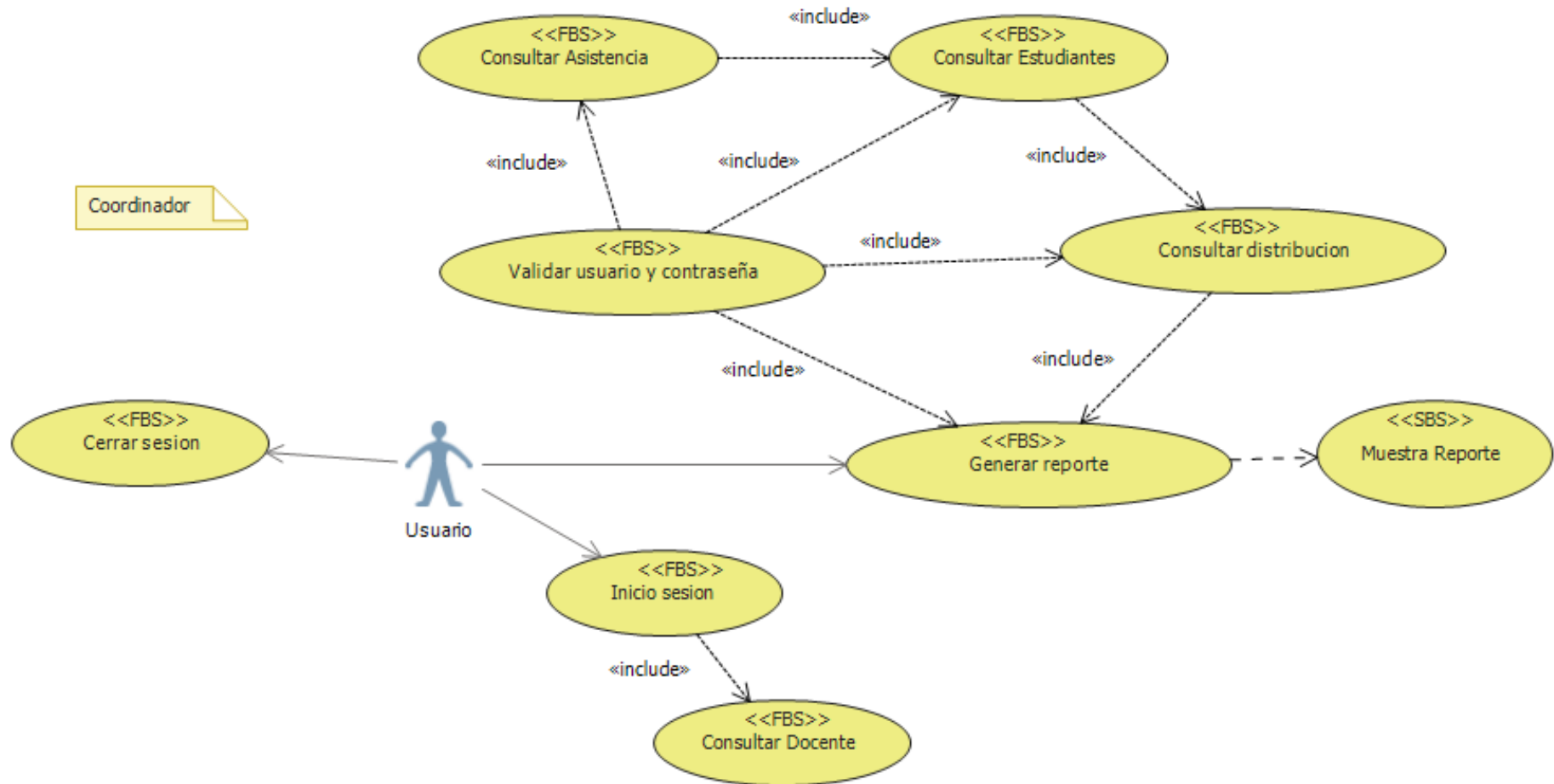


Figura 4.31. Modelo de Caso de Uso Extendido – Coordinador de Año

4.2.2.3.3. MODELO DE CASOS DE USOS EXTENDIDO- DIRECTOR DE CARRERA.

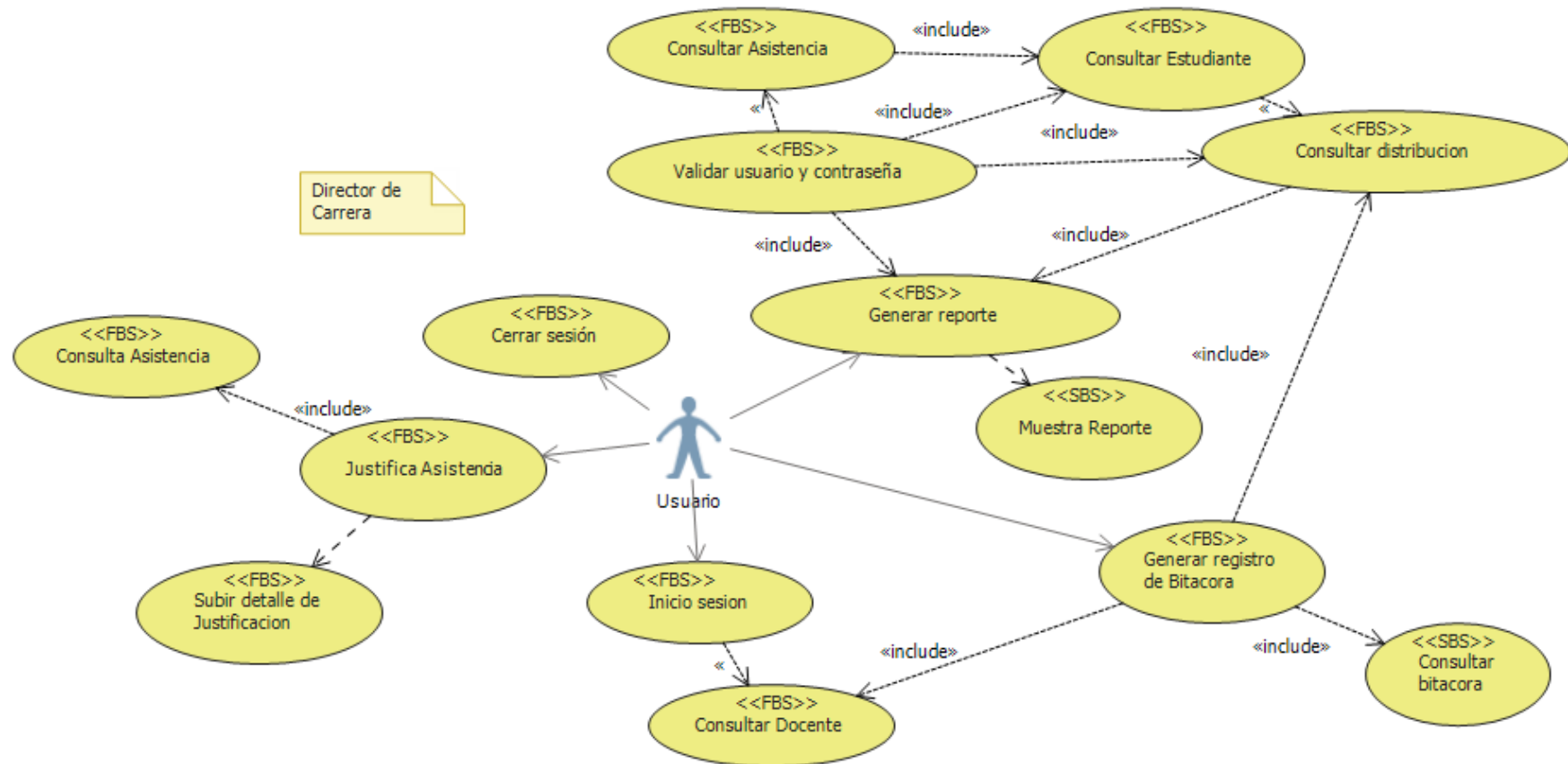


Figura 4.32. Modelo de Caso de Uso Extendido – Director de Carrera

4.2.2.3.4. MODELO DE CASOS DE USOS EXTENDIDO- DOCENTE.

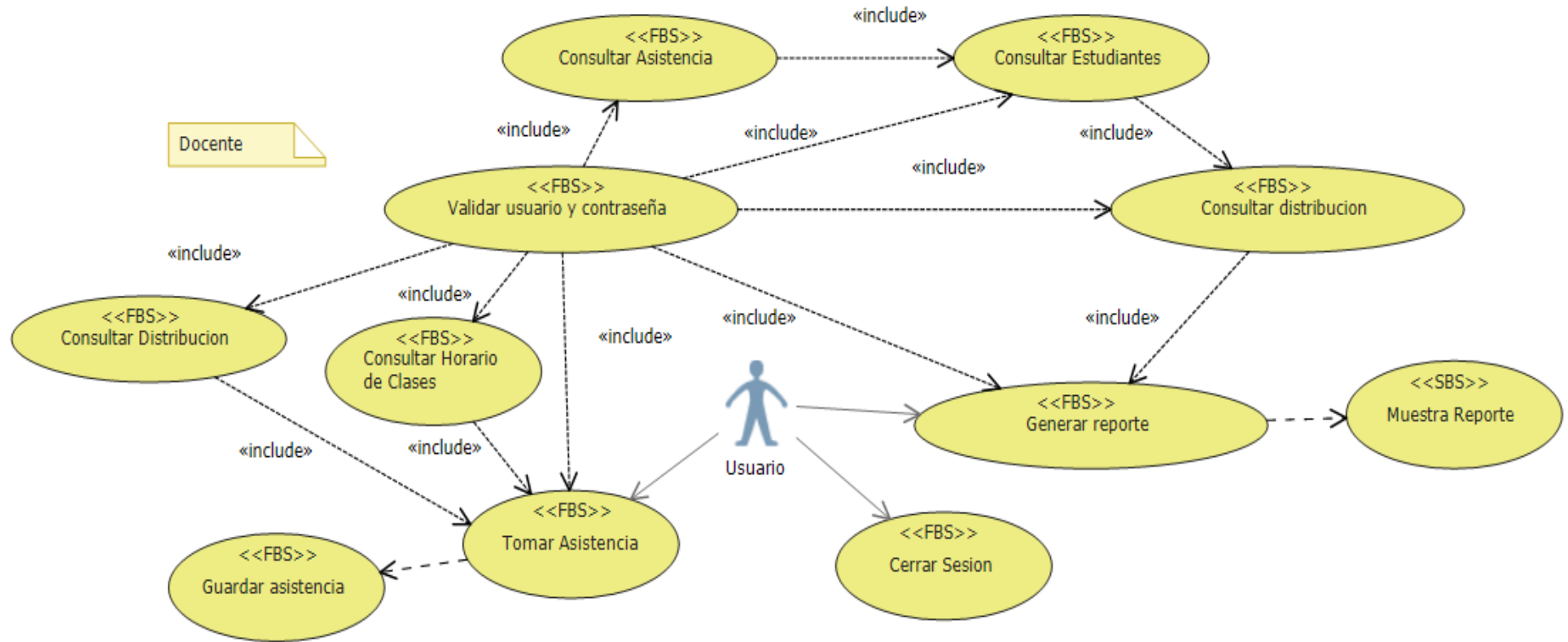


Figura 4.33. Modelo de Caso de Uso Extendido – Docente

4.2.2.3.5. MODELO DE CASOS DE USOS EXTENDIDO- ESTUDIANTE.

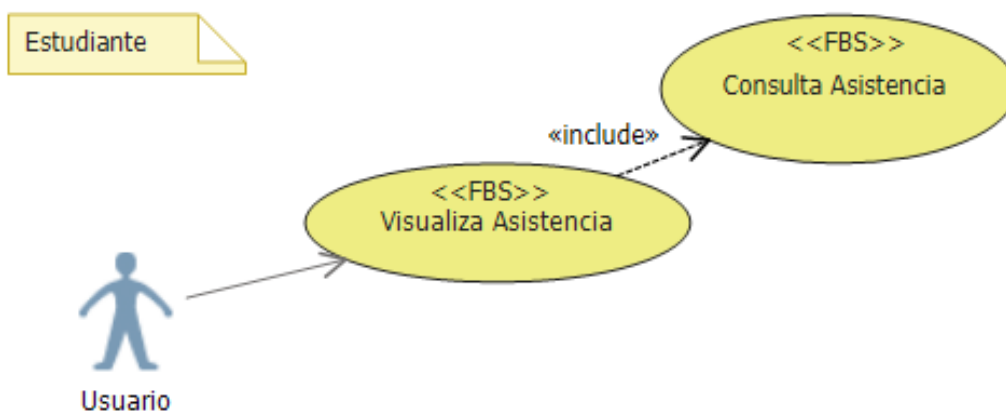


Figura 4.34. Modelo de Caso de Uso Extendido – Estudiante

4.2.2.3.6. MODELO DE CASOS DE USOS EXTENDIDO- ASISTENCIA.

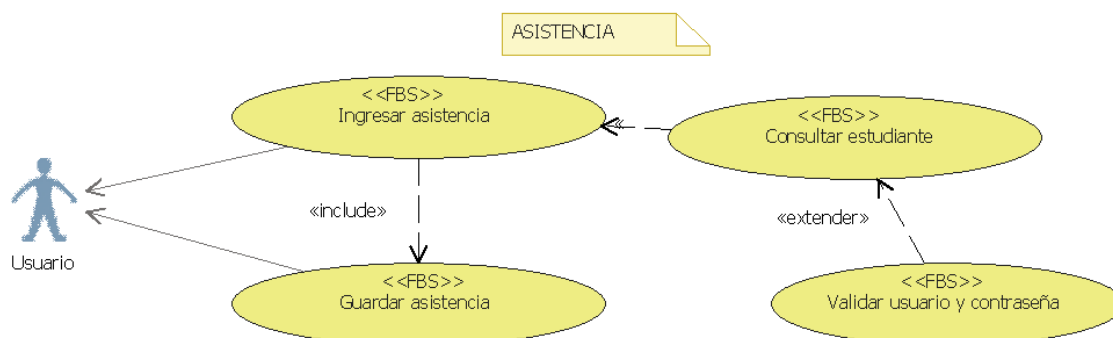


Figura 4.35. Modelo de Caso de Uso Extendido - Asistencia

4.2.2.3.7. MODELO DE CASOS DE USOS EXTENDIDO- CALENDARIO ACADÉMICO.

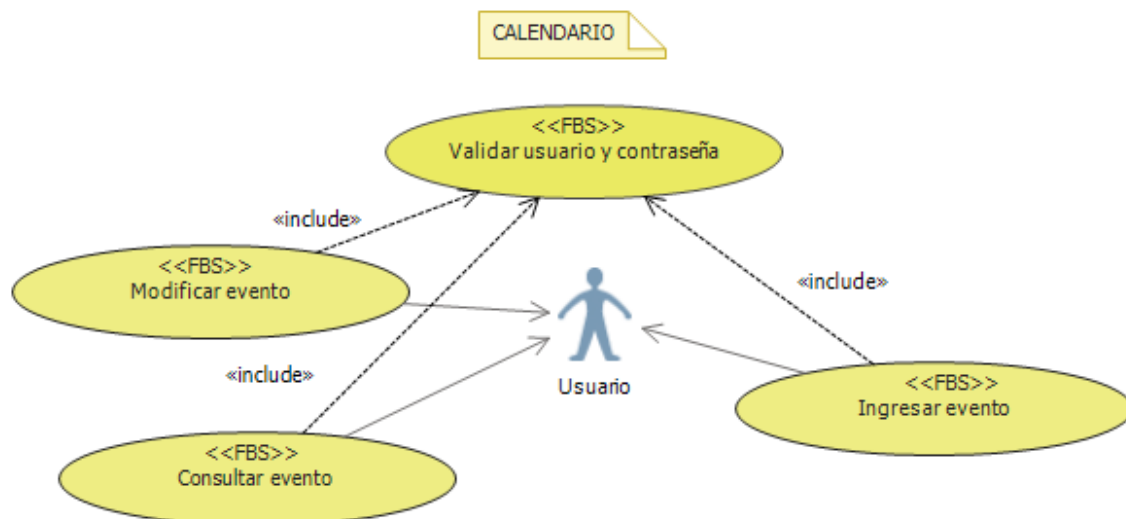


Figura 4.36. Modelo de Caso de Uso Extendido – Calendario Académico.

4.2.2.3.8. MODELO DE CASOS DE USOS EXTENDIDO- HORARIO DE CLASES.

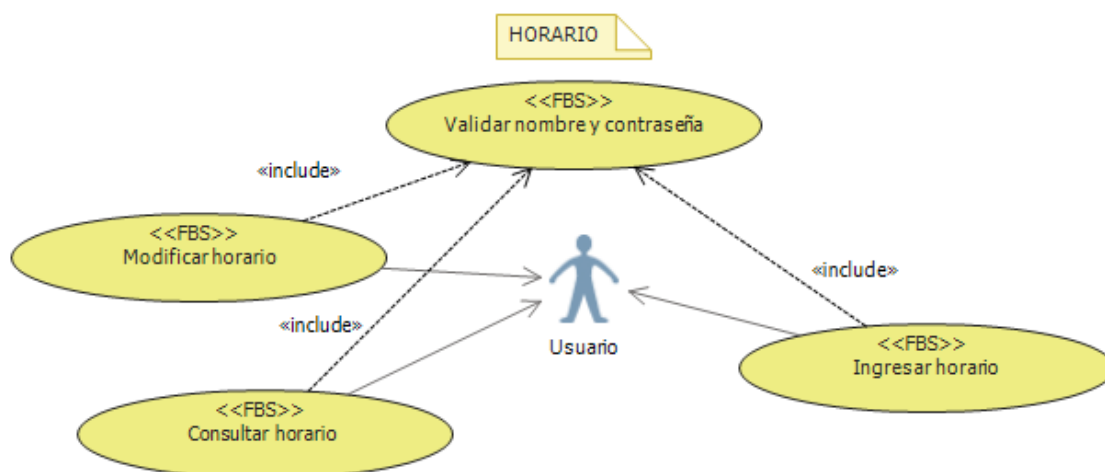


Figura 4.37. Modelo de Caso de Uso Extendido – Horario de Clases.

4.2.2.3.9. MODELO DE CASOS DE USOS EXTENDIDO- JUSTIFICACIÓN.

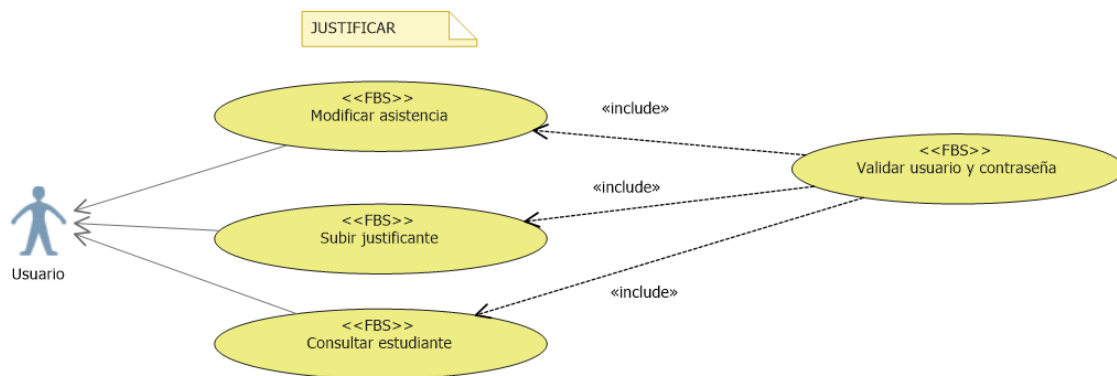


Figura 4.38. Modelo de Caso de Uso Extendido – Justificación.

4.2.2.3.10. MODELO DE CASOS DE USOS EXTENDIDO- REPORTE.

▸

Figura 4.39. Modelo de Caso de Uso Extendido – Reporte

4.2.2.4. MODELOS DE COMPOSICIÓN DE SERVICIOS

El modelo de composición de servicios mostró una visión clara de cómo los servicios están relacionados entre sí, las interacciones que realizan los procesos y que acciones realizan cada módulo en el sistema.

4.2.2.4.1. MODULO DE ASISTENCIA

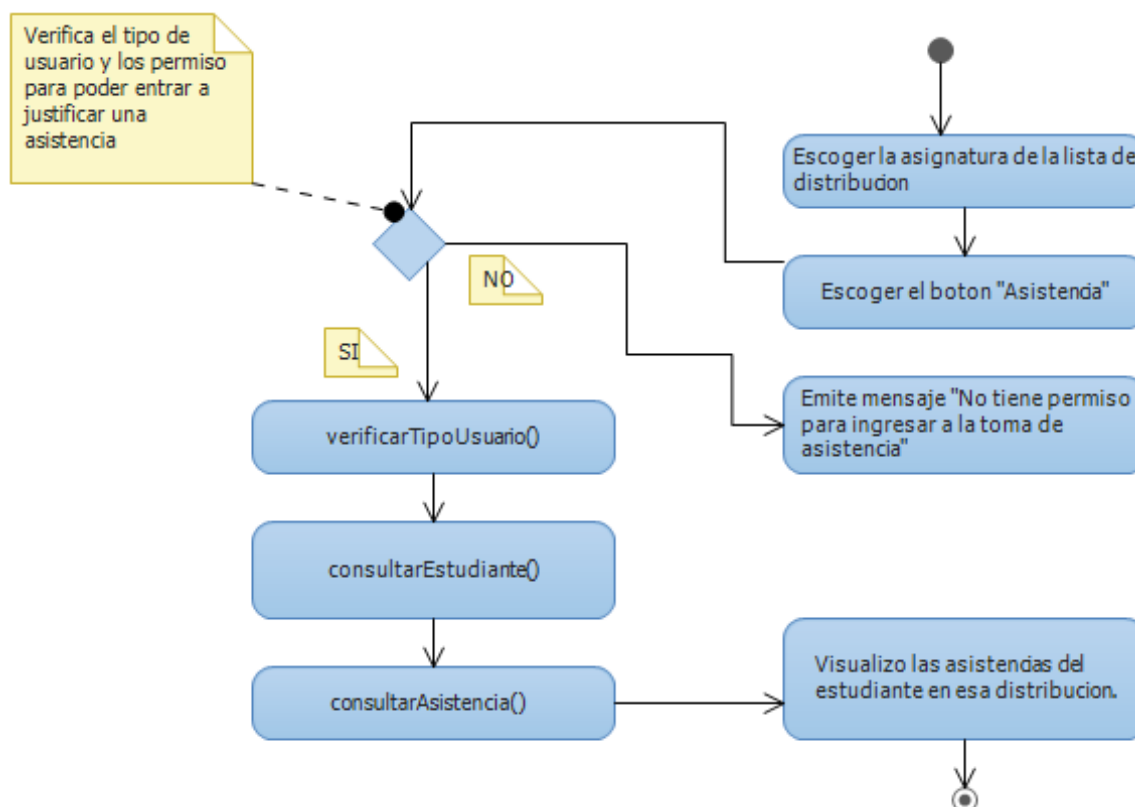


Figura 4.40. Modelo de Composición de Servicio - Consultar Cabecera

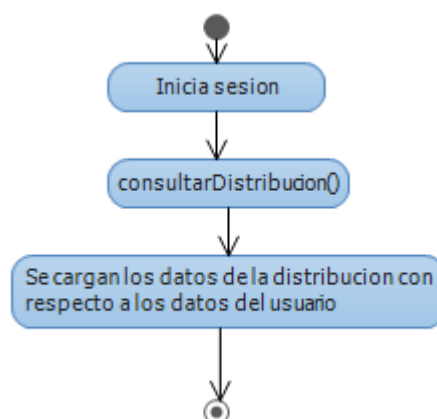


Figura 4.41. Modelo de Composición de Servicio - Consultar Distribución

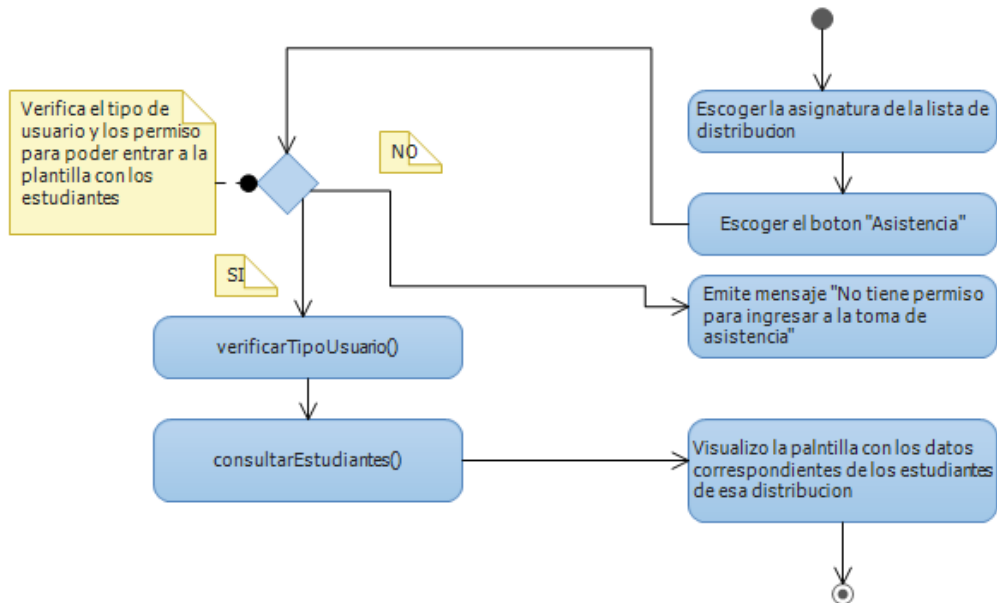


Figura 4.42. Modelo de Composición de Servicio - Consultar Estudiante

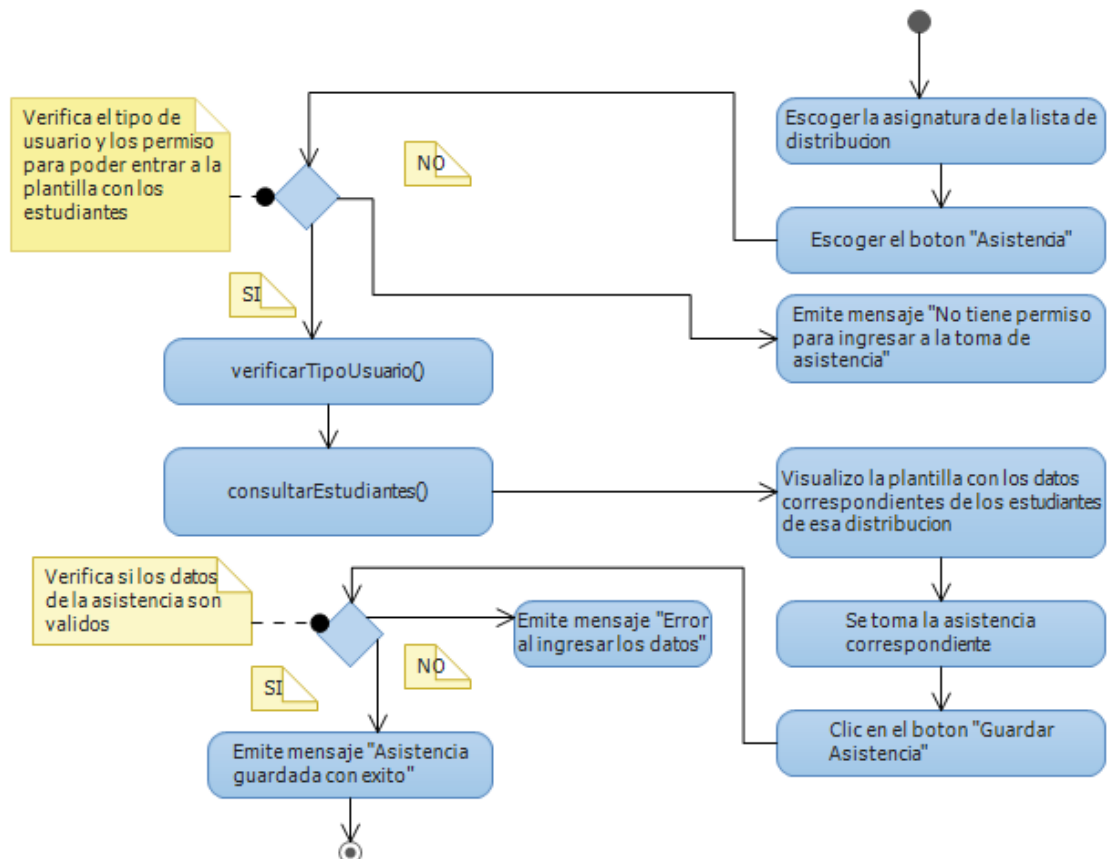


Figura 4.43. Modelo de Composición de Servicio - Guardar Asistencia

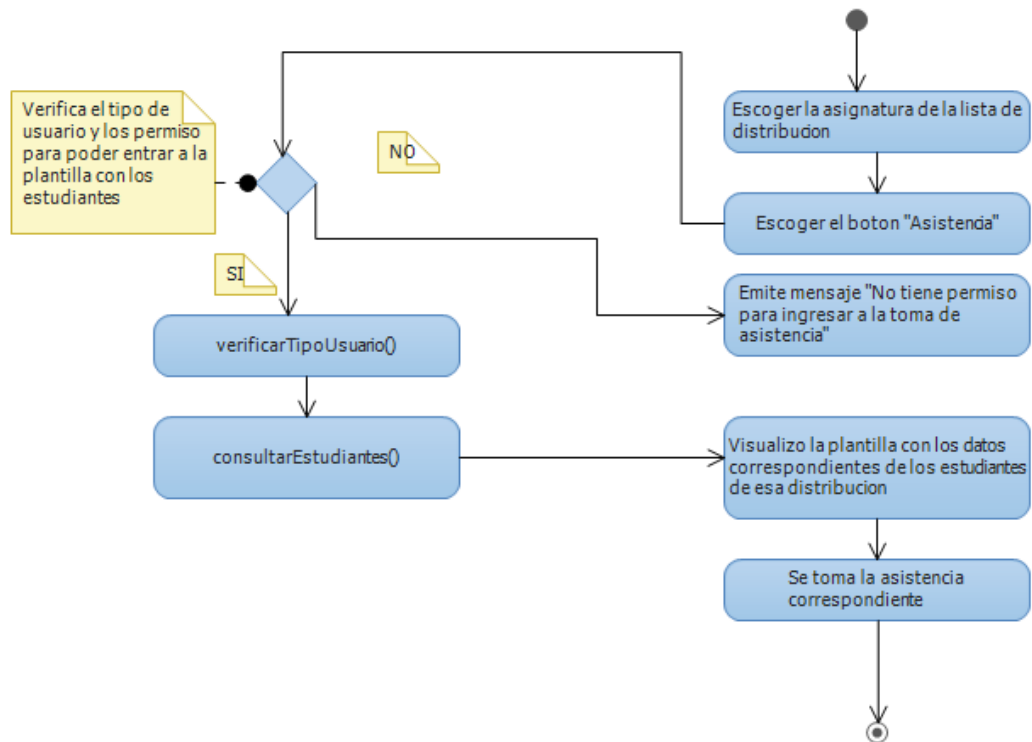


Figura 4.44. Modelo de Composición de Servicio - Ingresar Asistencia

4.2.2.4.2. MODULO DE CALENDARIO ACADÉMICO

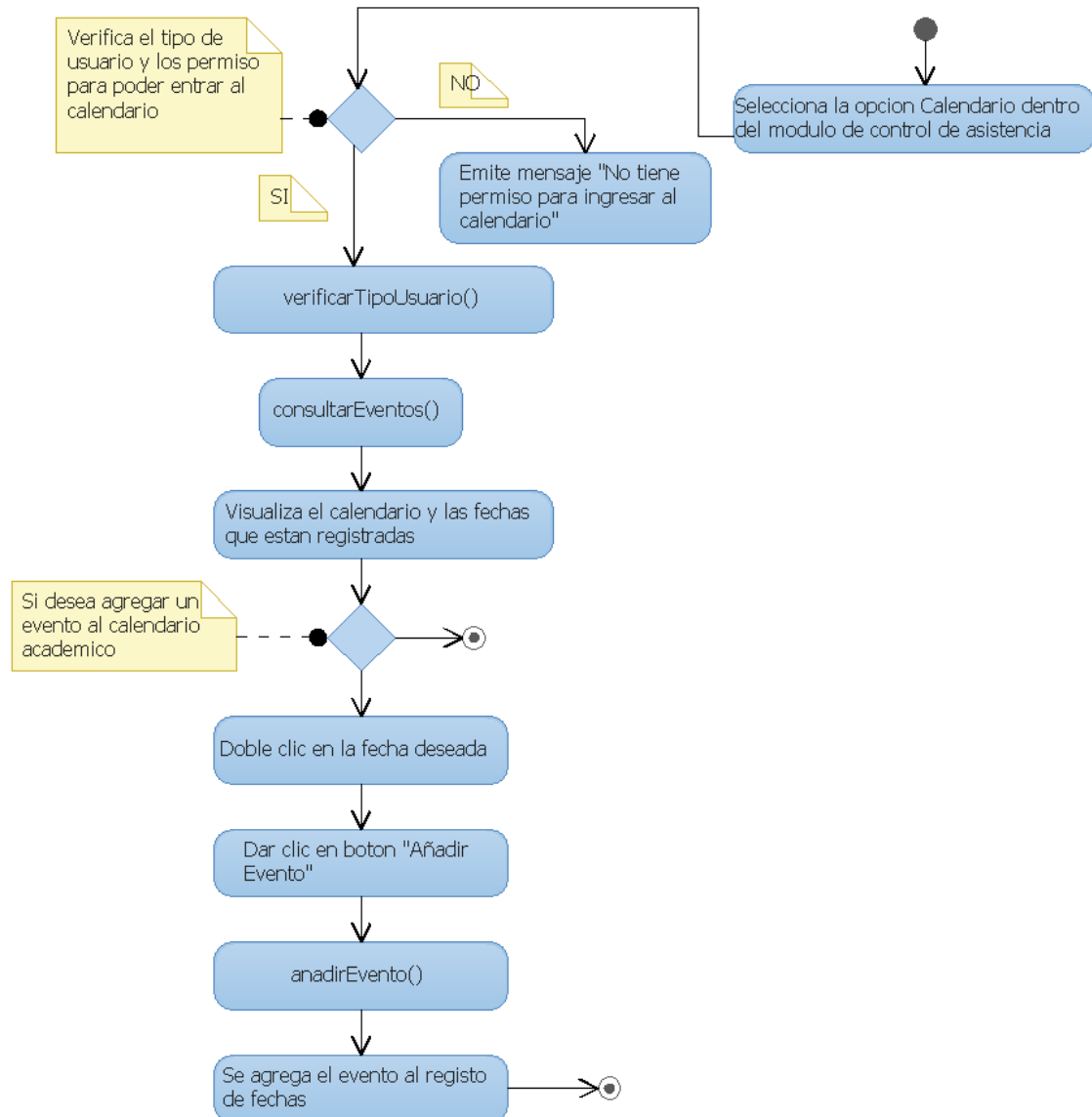


Figura 4.45. Modelo de Composición de Servicio – Añadir Evento

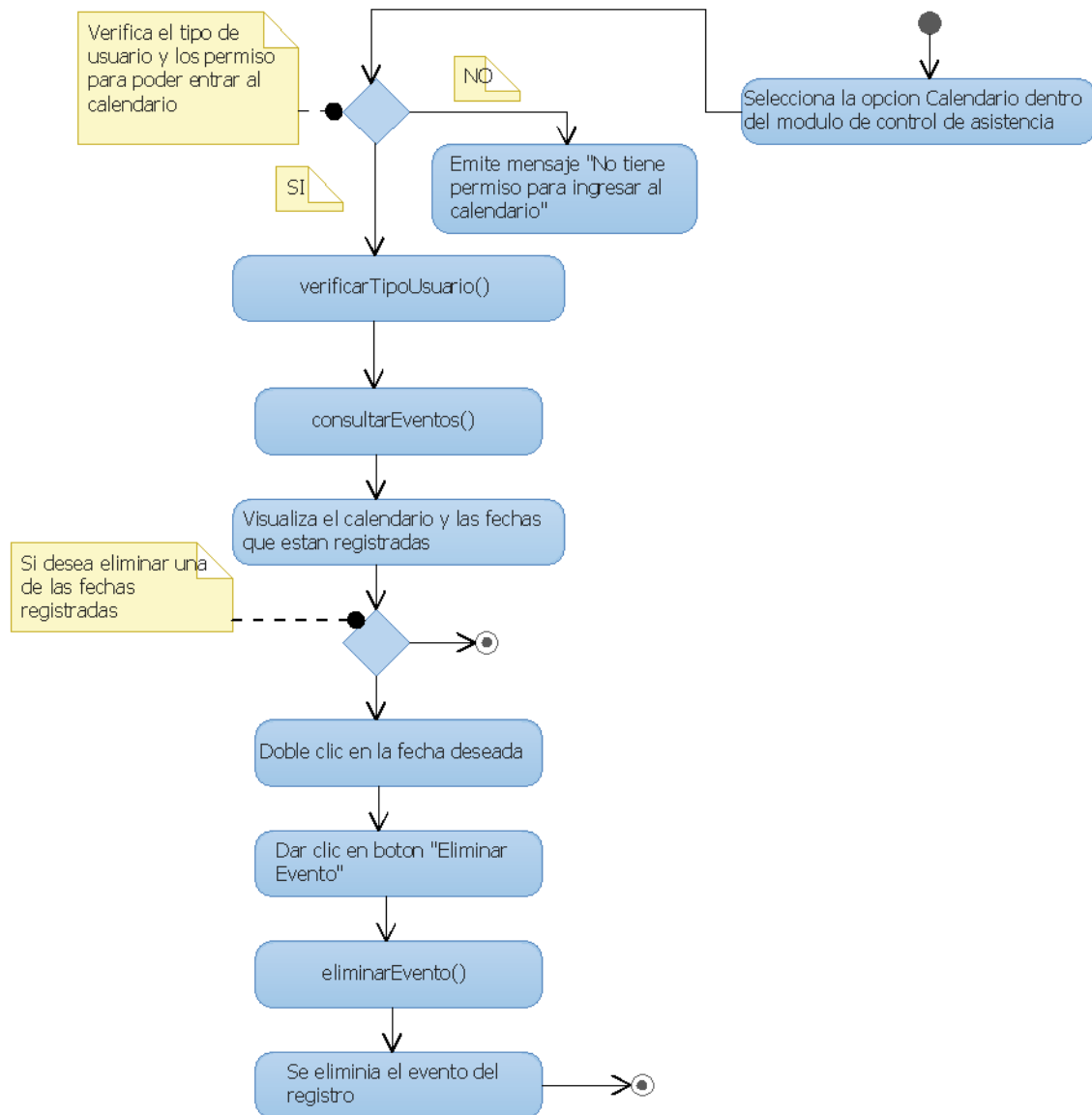


Figura 4.46. Modelo de Composición de Servicio – Eliminar Evento

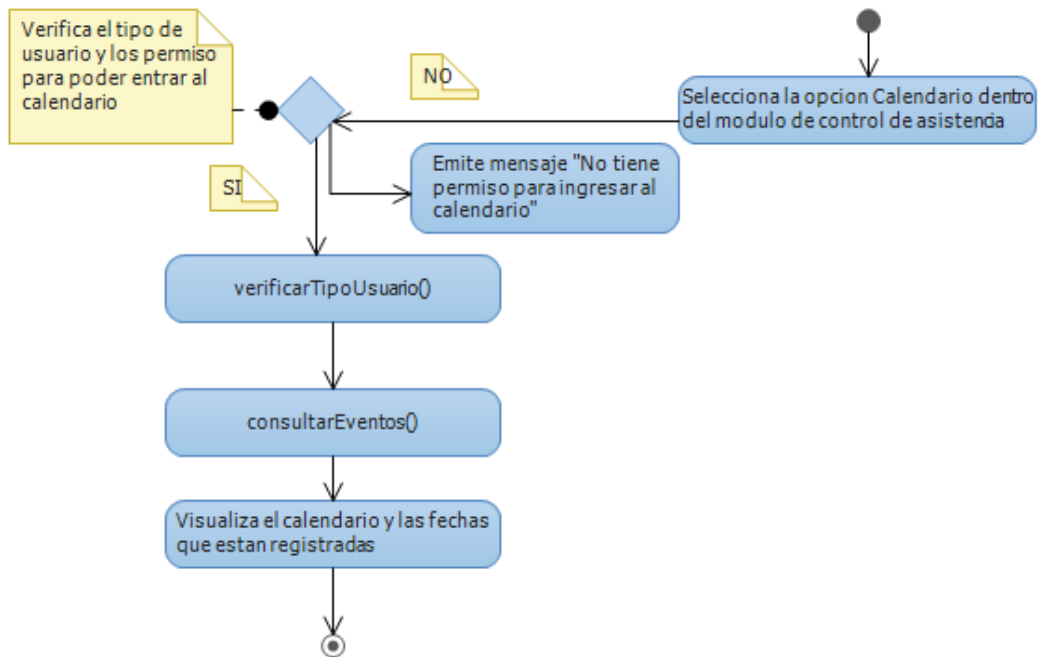


Figura 4.47. Modelo de Composición de Servicio – Consultar Eventos

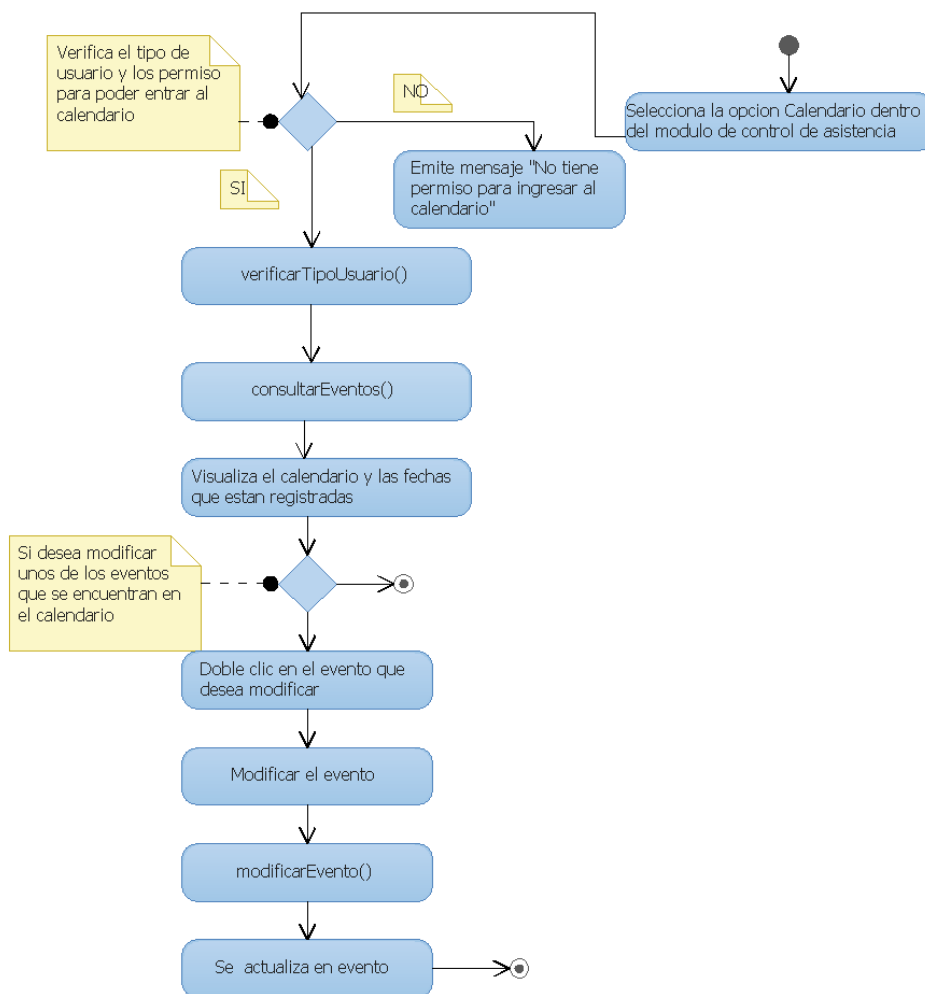


Figura 4.48. Modelo de Composición de Servicio – Modificar Evento.

4.2.2.4.3. MODULO DE HORARIO DE CLASES

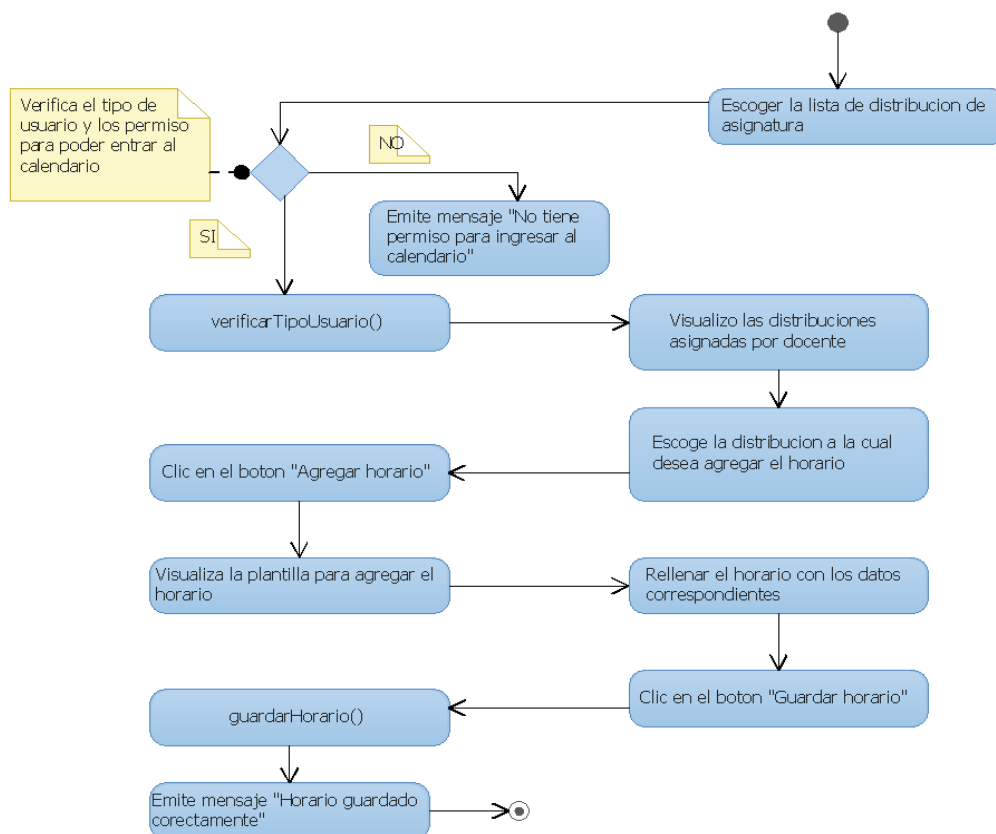


Figura 4.49. Modelo de Composición de Servicio – Añadir Horario.

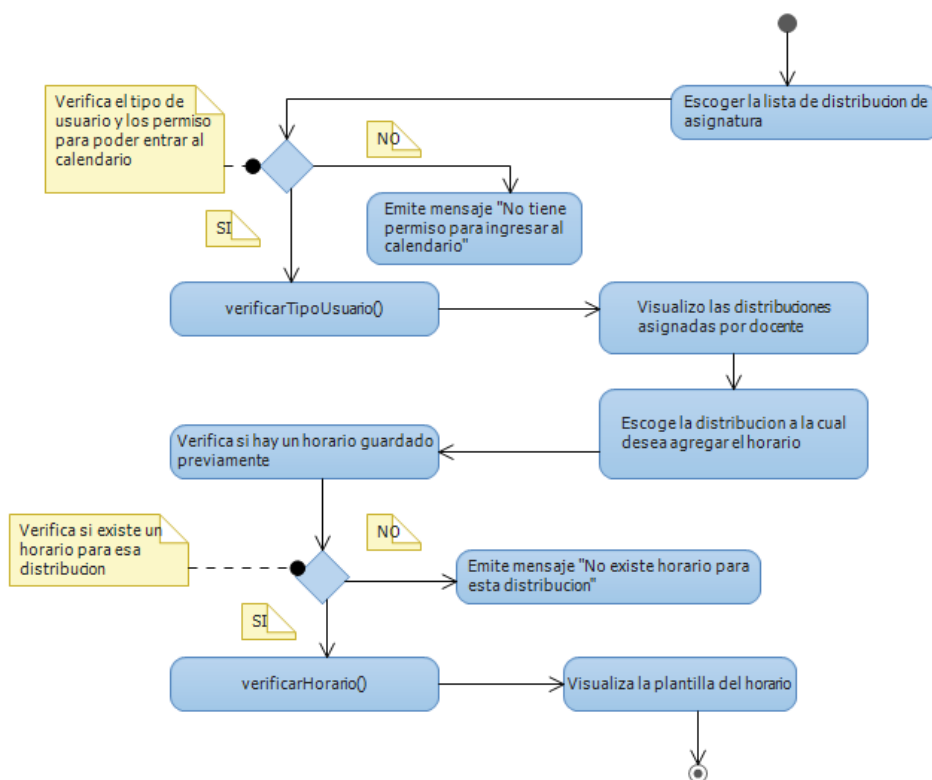


Figura 4.50. Modelo de Composición de Servicio – Consultar Horario.

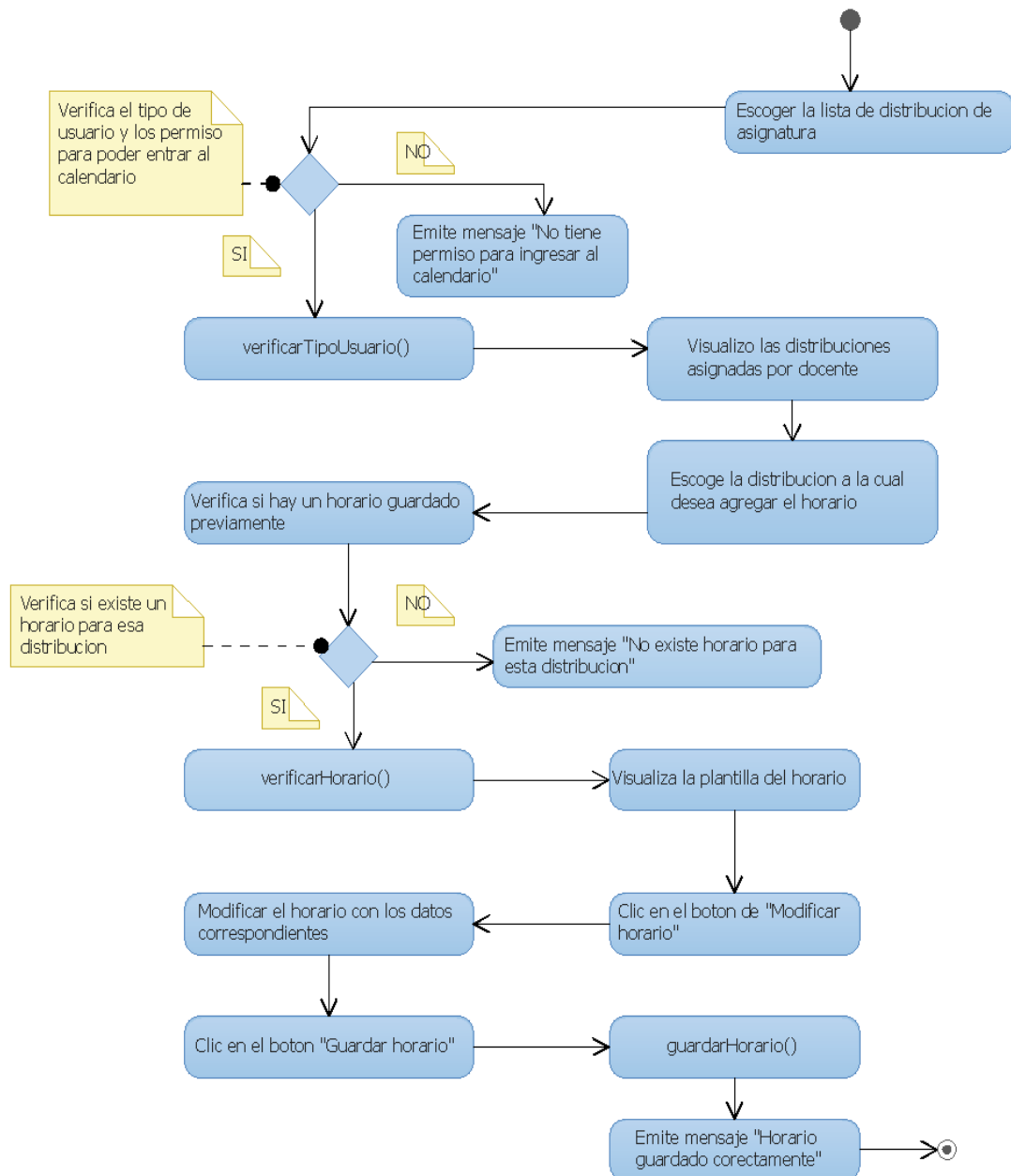


Figura 4.51. Modelo de Composición de Servicio – Modificar Horario

4.2.2.4.4. MODULO DE INGRESO AL SISTEMA

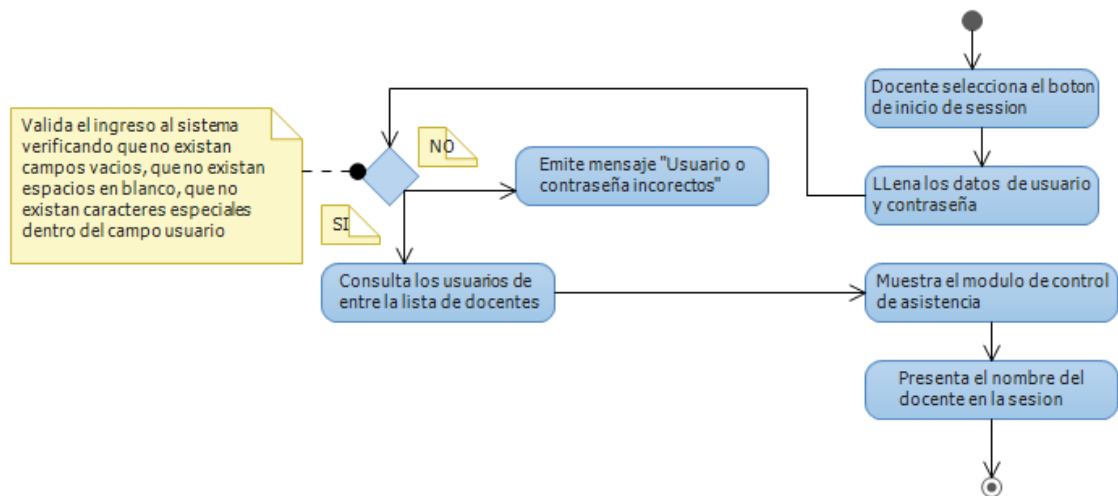


Figura 4.52. Modelo de Composición de Servicio – Consultar Docente.

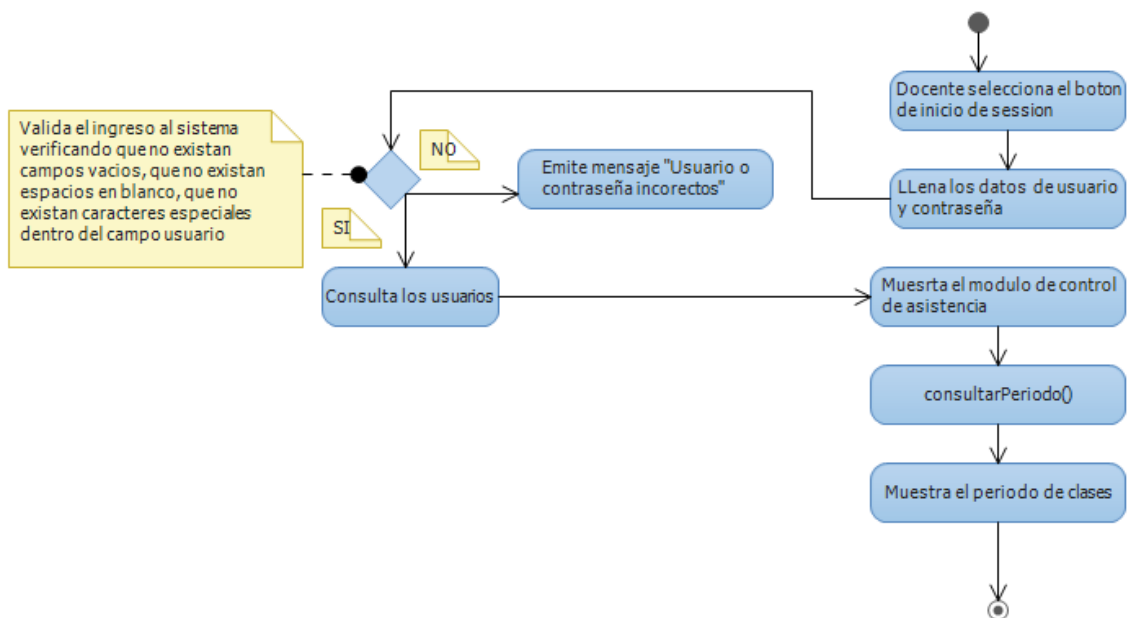


Figura 4.53. Modelo de Composición de Servicio – Consultar Periodo.

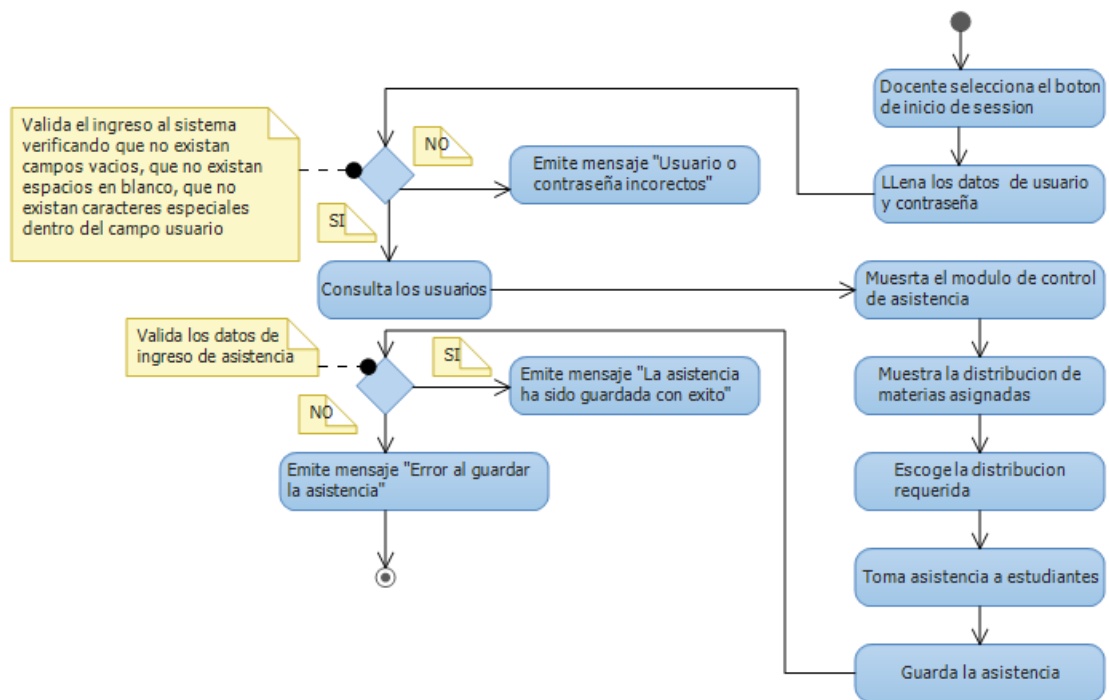


Figura 4.54. Modelo de Composición de Servicio – Iniciar Sesión.

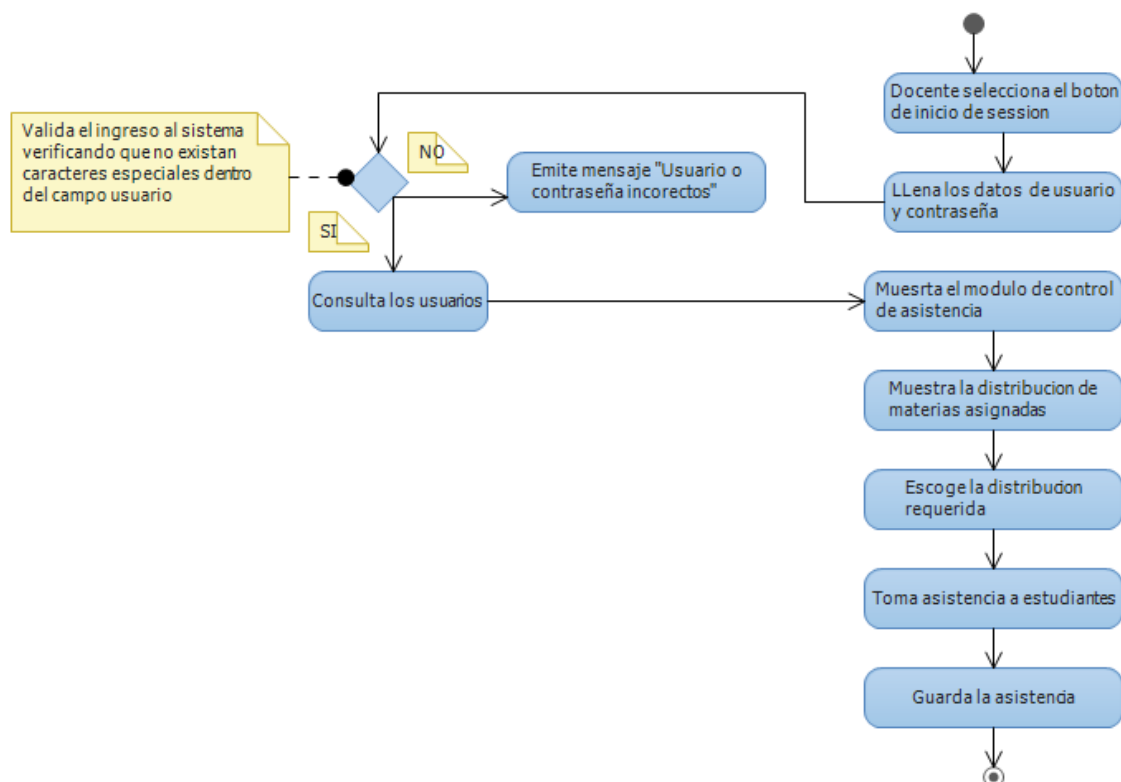


Figura 4.55. Modelo de Composición de Servicio – Validar Caracteres Especiales.

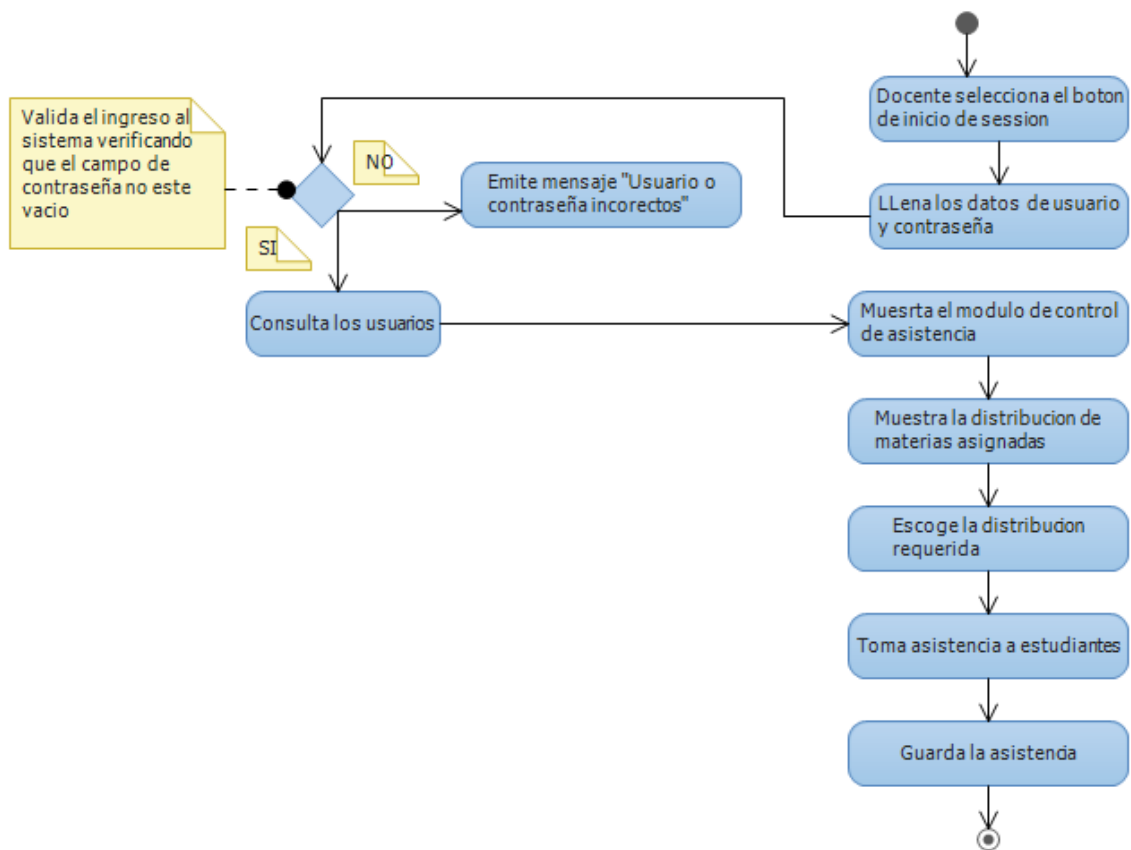


Figura 4.56. Modelo de Composición de Servicio – Validar Campo Contraseña

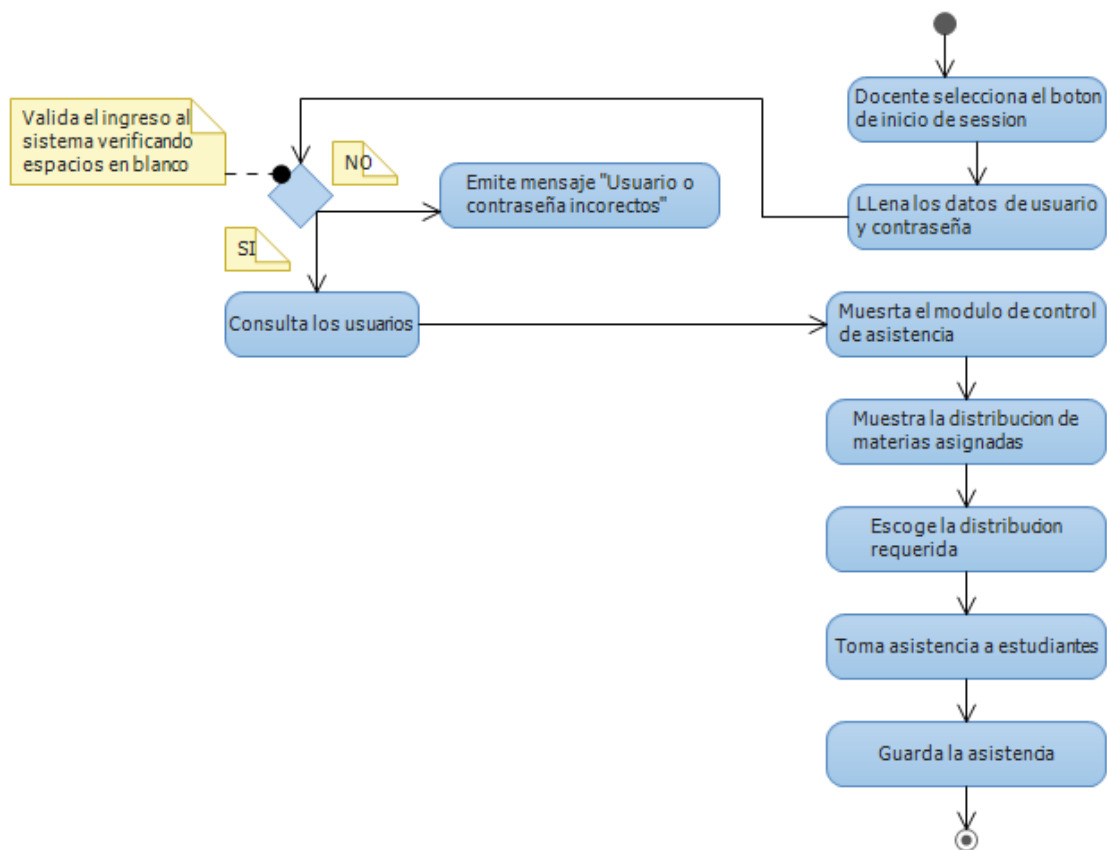


Figura 4.57. Modelo de Composición de Servicio – Validar Espacio en Blanco

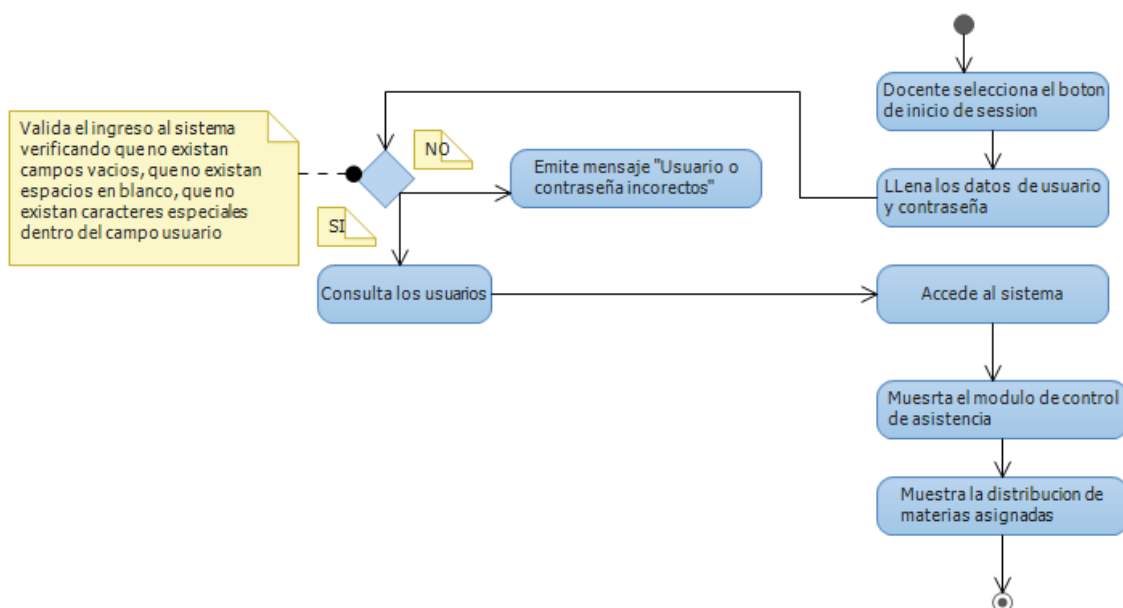


Figura 4.58. Modelo de Composición de Servicio – Validar Perfil Usuario

4.2.2.4.5. MODULO DE JUSTIFICAR ASISTENCIA.

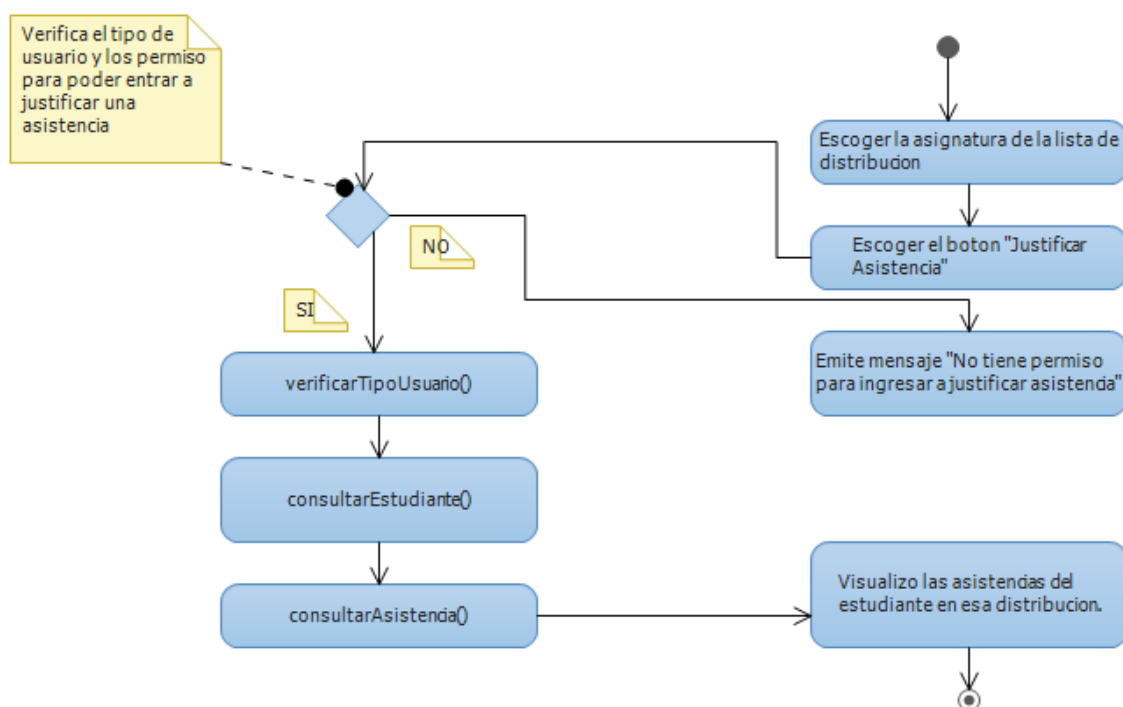


Figura 4.59. Modelo de Composición de Servicio – Consultar Asistencia.

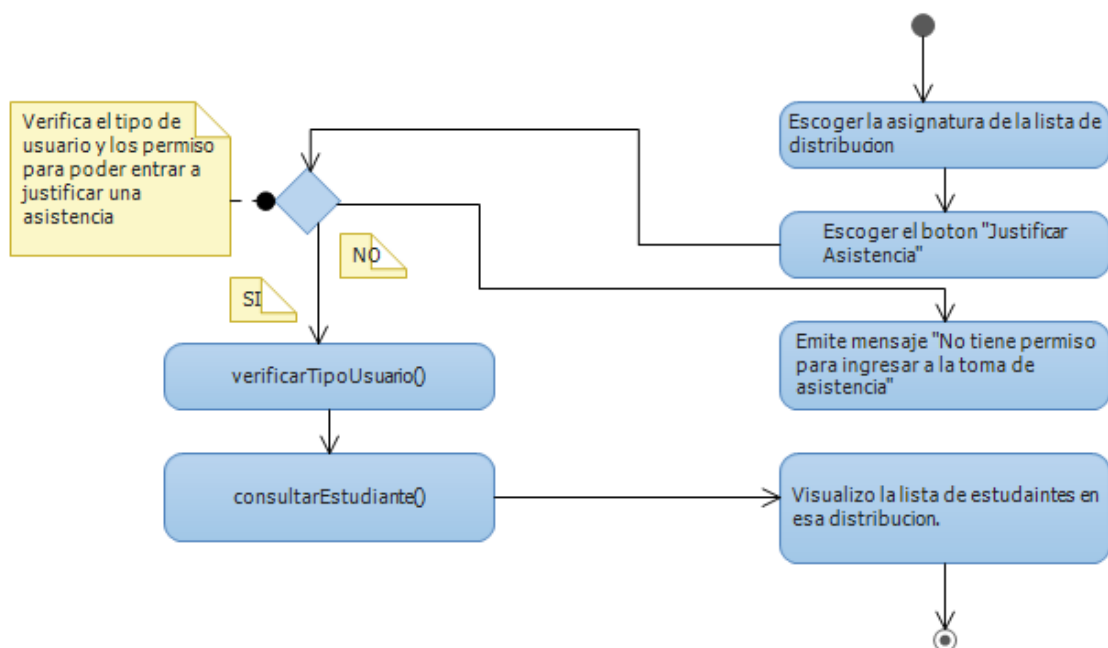


Figura 4.60. Modelo de Composición de Servicio – Consultar Estudiante.

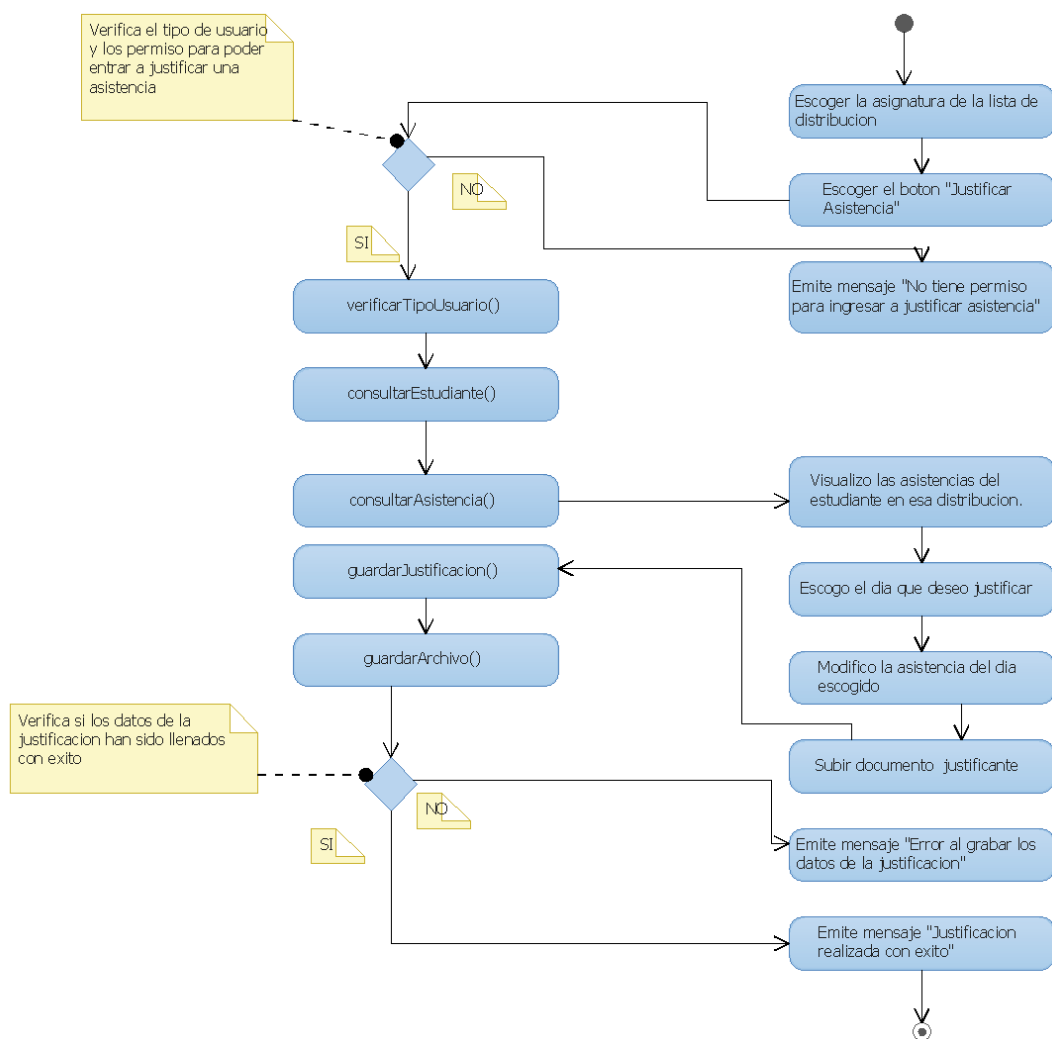


Figura 4.61. Modelo de Composición de Servicio – Ingresar Justificación

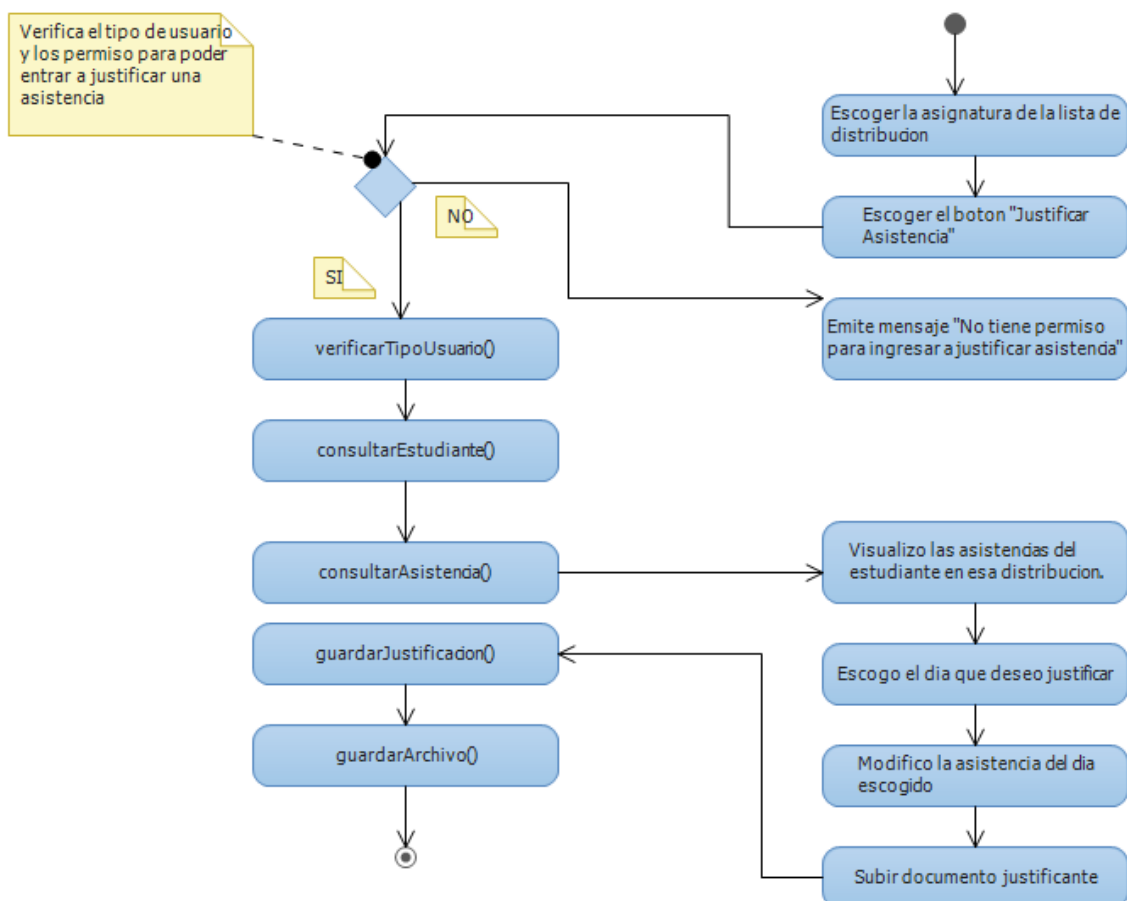


Figura 4.62. Modelo de Composición de Servicio –Modificar Asistencia

4.2.2.4.6. MODULO DE REPORTE.

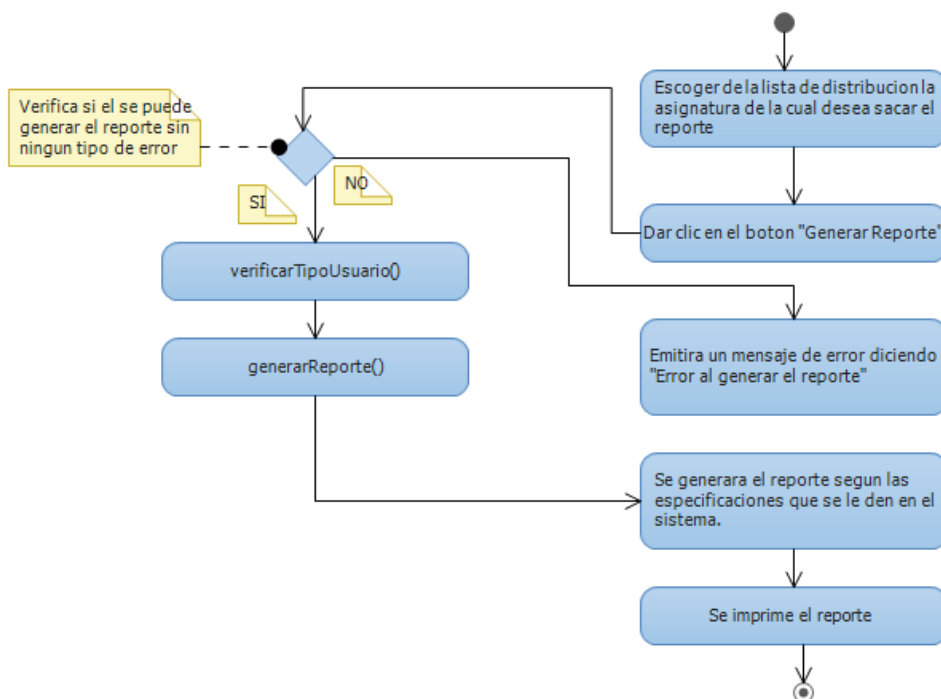


Figura 4.63. Modelo de Composición de Servicio – Generar Reporte.

4.2.2.4.7. MODULO DE BITÁCORA.

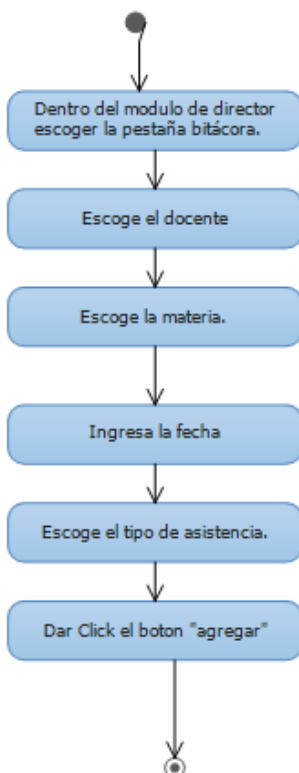


Figura 4.64. Modelo de Composición de Servicio –Agregar bitácora

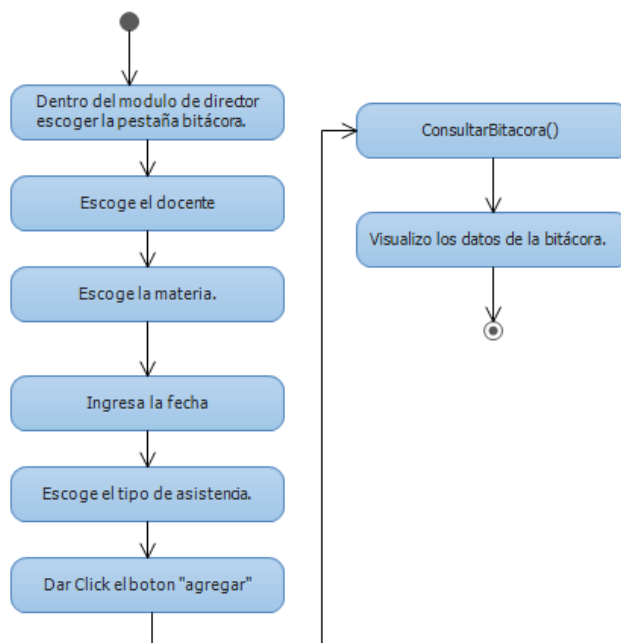


Figura 4.65. Modelo de Composición de Servicio –Consultar bitácora

4.2.2.4.8. MODULO DE TIEMPO

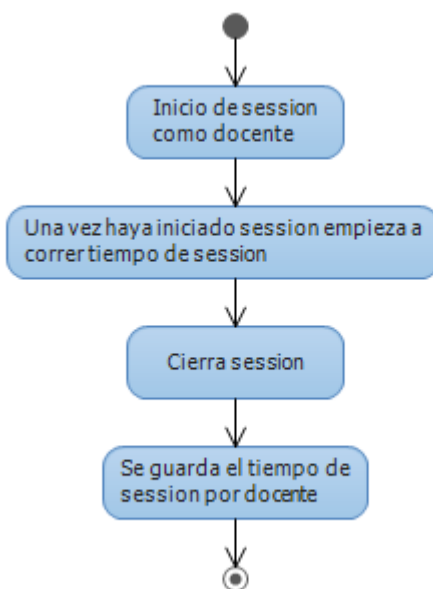


Figura 4.66. Modelo de Composición de Servicio –Agregar tiempo

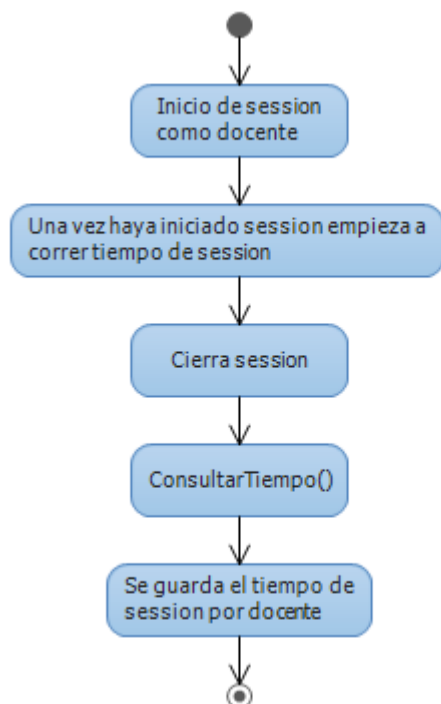


Figura 4.67. Modelo de Composición de Servicio –Consultar tiempo

4.2.2.4.9. MODULO DE COORDINADOR

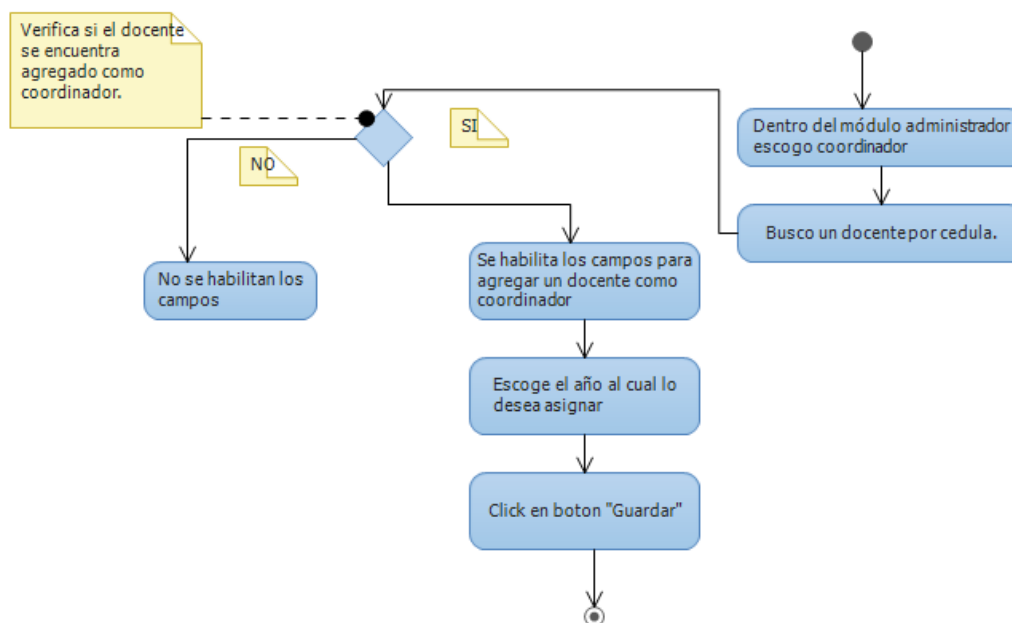


Figura 4.68. Modelo de Composición de Servicio –Agregar coordinador.

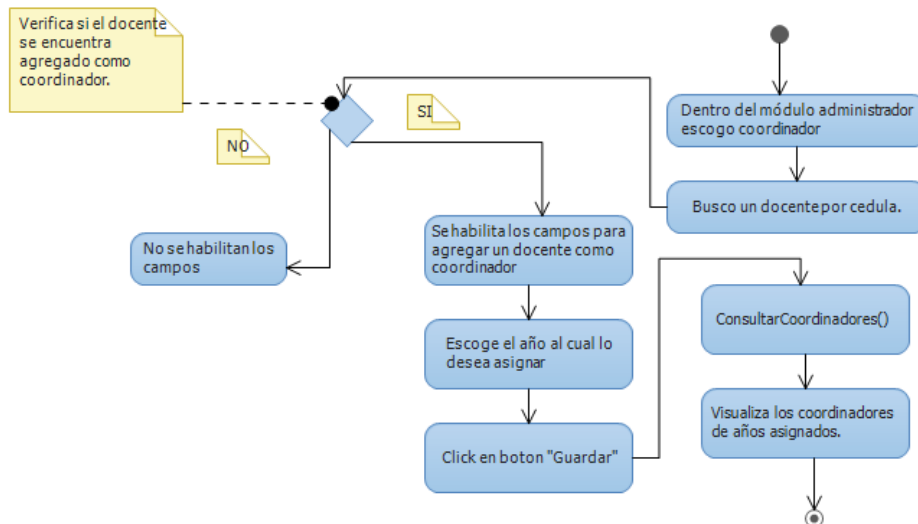


Figura 4.69. Modelo de Composición de Servicio –Consultar coordinador.

4.2.2.4.10. MODULO DE PERSONALIZACIÓN.

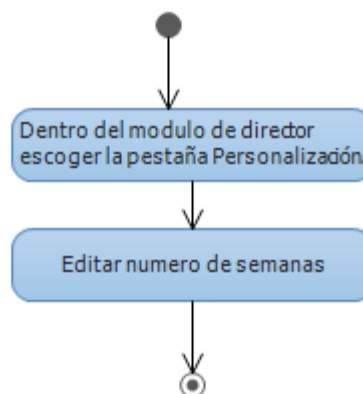


Figura 4.70. Modelo de Composición de Servicio –Agregar Personalización.

4.2.2.5. MODELO DE FRAGMENTOS

El modelo de fragmentos, muestra la estructura interna de las funciones que tiene el aplicativo, describiendo si las funciones son fragmentos estructurales (SS), es decir que muestran información, o son fragmentos estructurales (FS), funciones que necesitan parámetros para realizar algún tipo de proceso.



Figura 4.71. Modelo de Fragmentos Estructurales

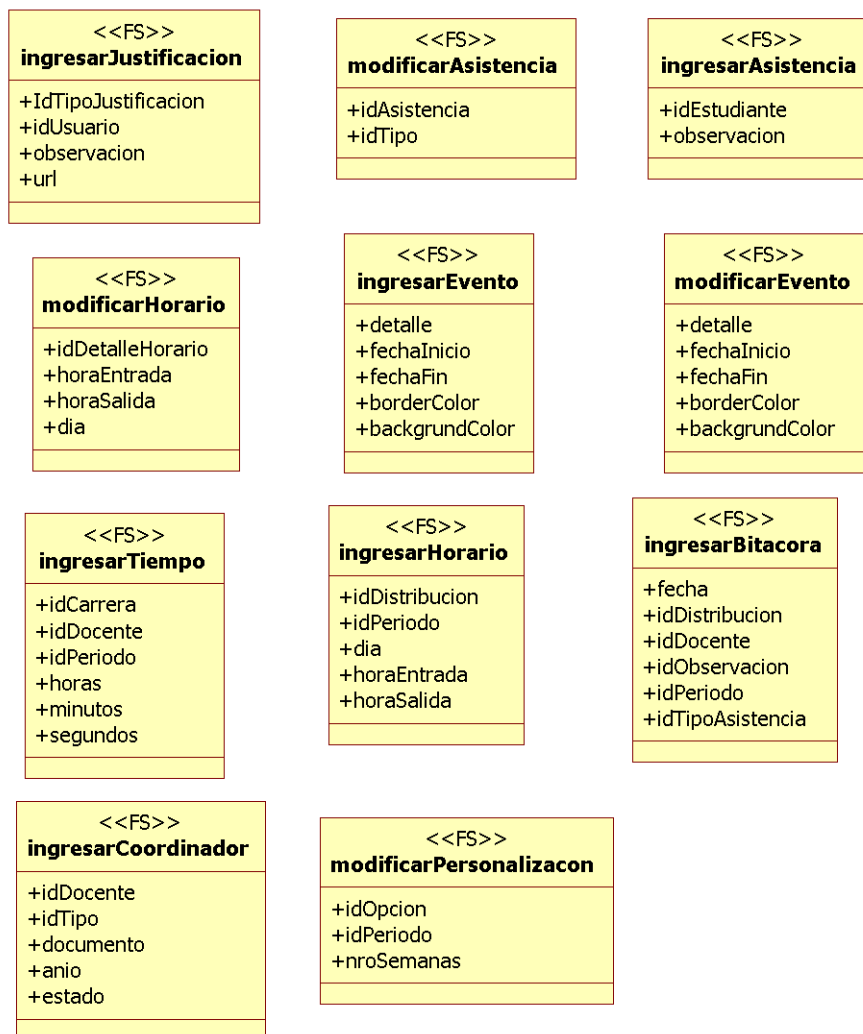


Figura 4.72. Modelo de Fragmentos Funcionales

4.2.2.6. MODELO DE NAVEGACIÓN

El Modelo de Navegación mostró la distribución de las funciones, en los diferentes modulo del aplicativo web, mediante un diagrama de estructura de que se lo realizo.

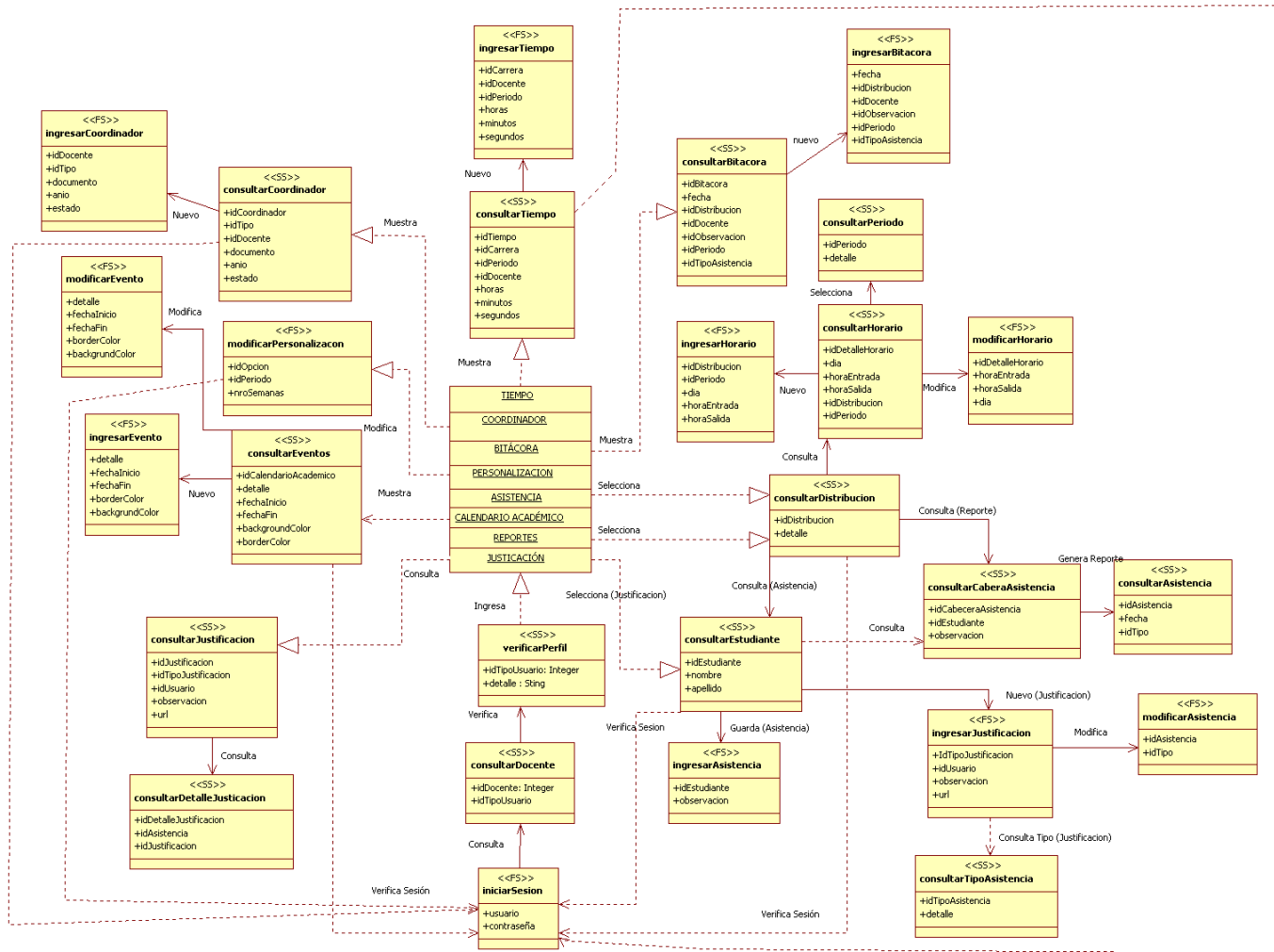


Figura 4.73. Modelo de Navegación

4.2.2.7. MODELO DE SERVICIOS

Los servicios conceptuales que ofrece el sistema de información Web están descritos en el Modelo de Servicio, mostrando cada servicio que tiene la aplicación que mediante un diagrama de comunicación, describió la interacción que se mantiene las unidades de información, expresando las dependencias de las funciones a los objetos.

4.2.2.7.1. MODULO DE ASISTENCIA

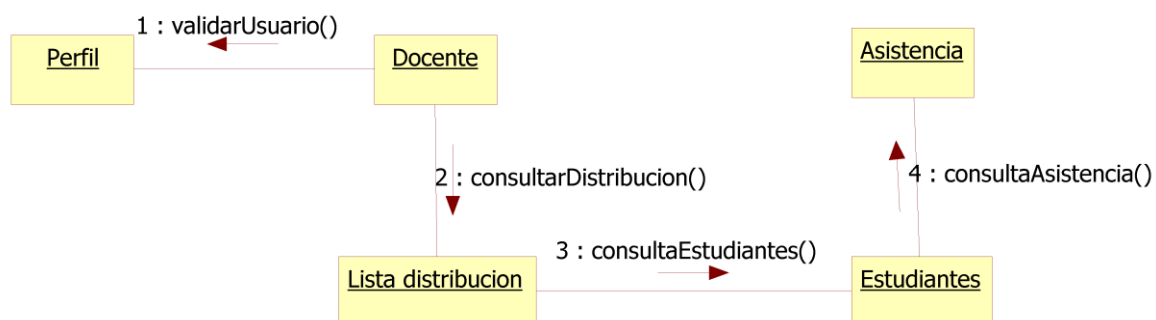


Figura 4.74. Modelo de Servicio – Consultar Asistencia

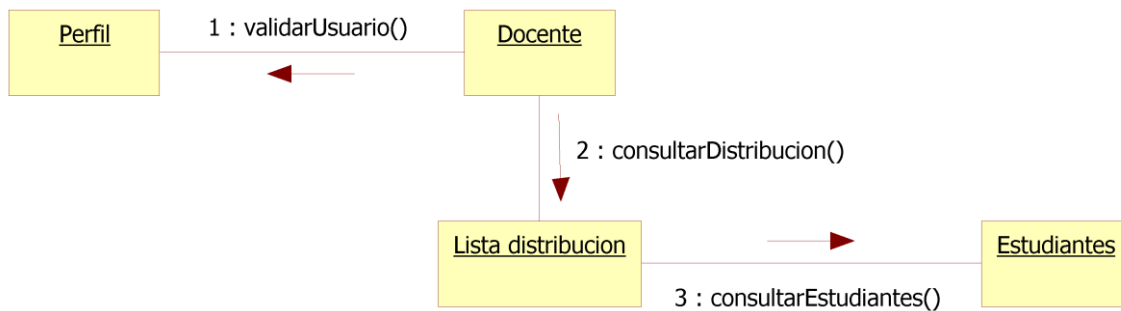


Figura 4.75. Modelo de Servicio – Consulta Estudiante

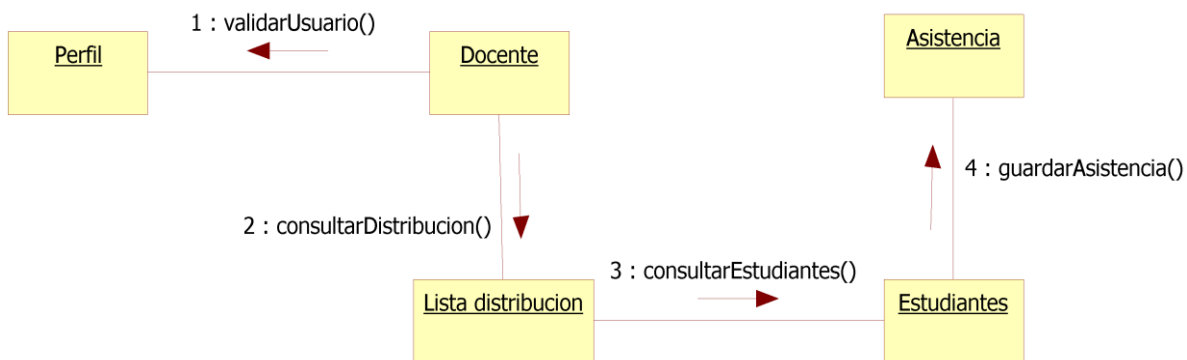


Figura 4.76. Modelo de Servicio – Guardar Asistencia

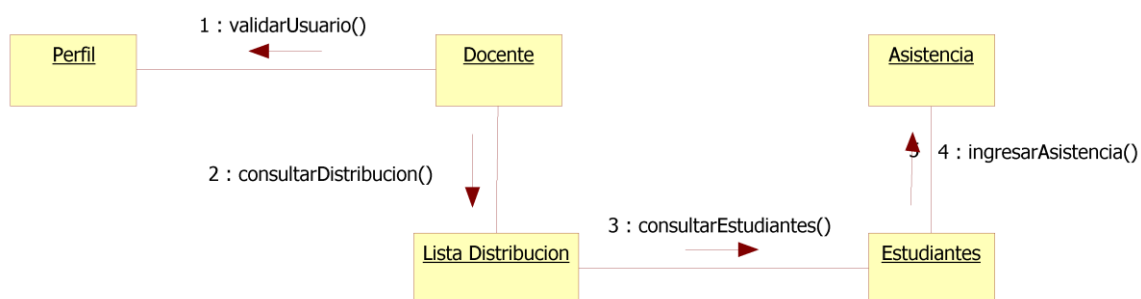


Figura 4.77. Modelo de Servicio – Ingresar Asistencia.

4.2.2.7.2. MODULO DE CALENDARIO ACADÉMICO

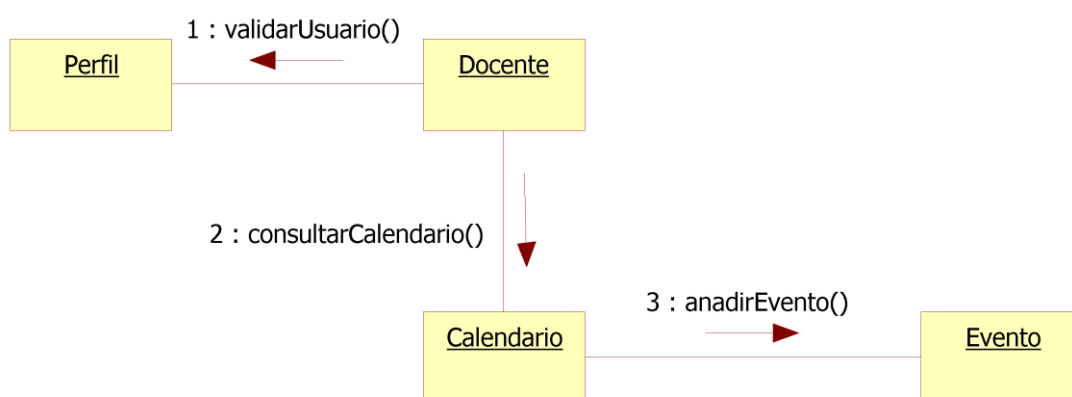


Figura 4.78. Modelo de Servicio – Añadir Evento

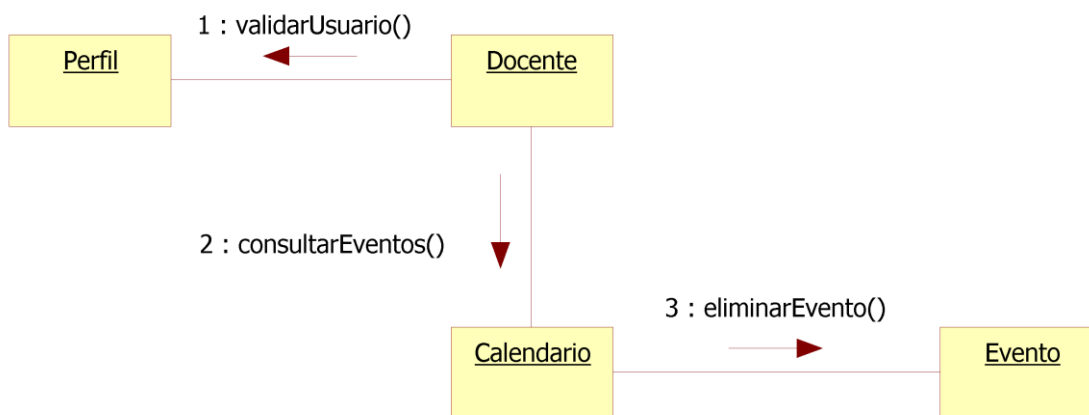


Figura 4.79. Modelo de Servicio – Eliminar Evento.

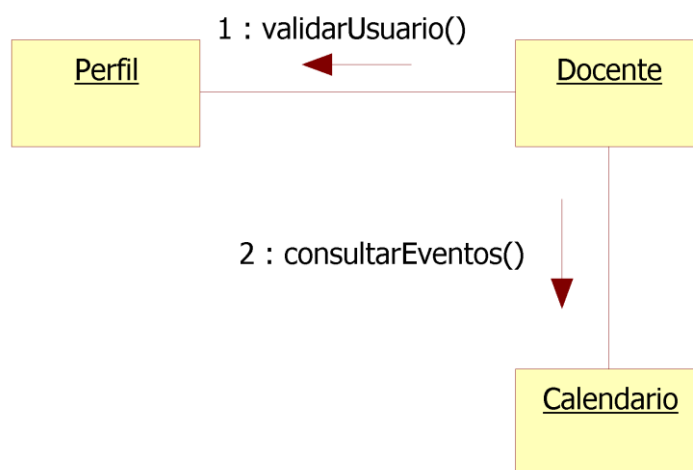


Figura 4.80. Modelo de Servicio – Consultar Evento.

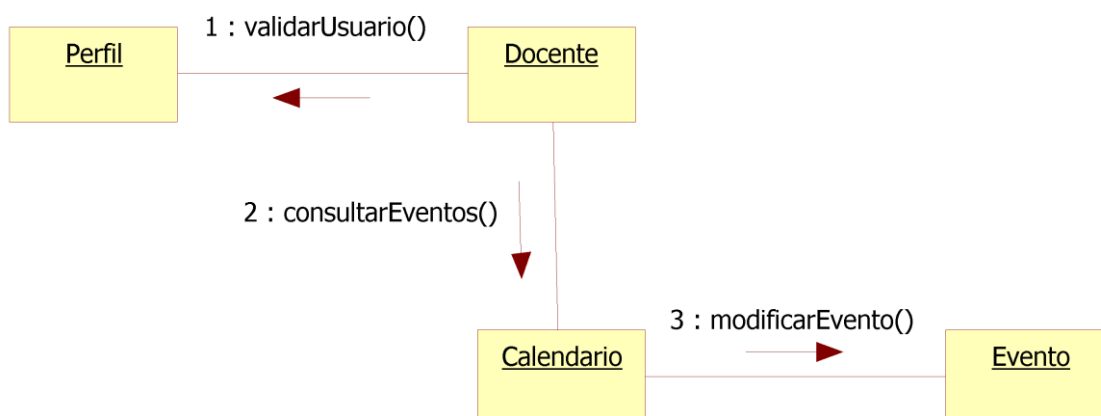


Figura 4.81. Modelo de Servicio – Modificar Evento.

4.2.2.7.3. MODULO DE HORARIO DE CLASES

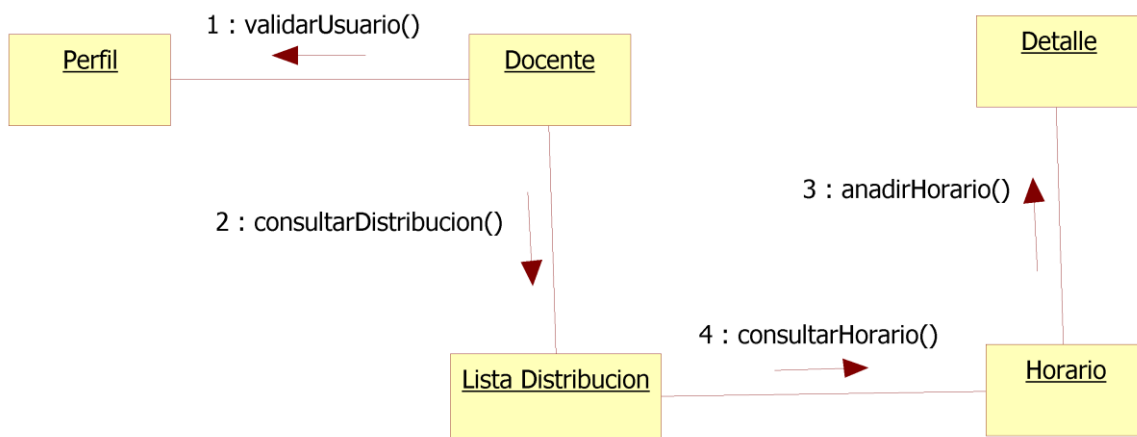


Figura 4.82. Modelo de Servicio – Añadir Horario.

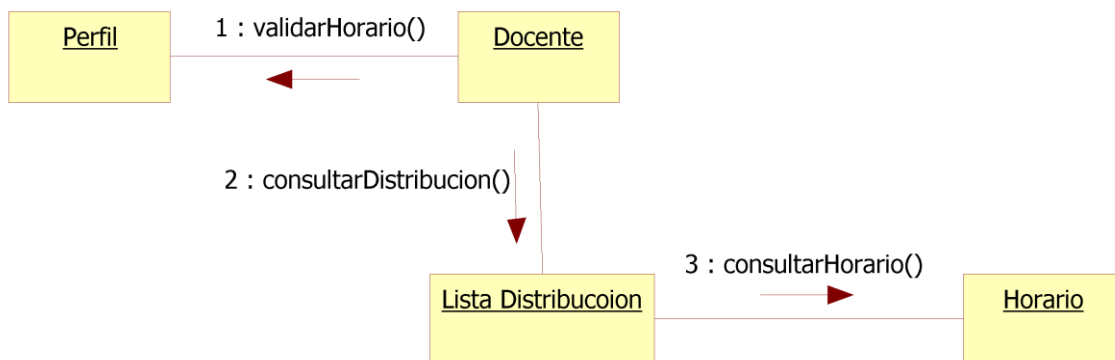


Figura 4.83. Modelo de Servicio – Consultar Horario.

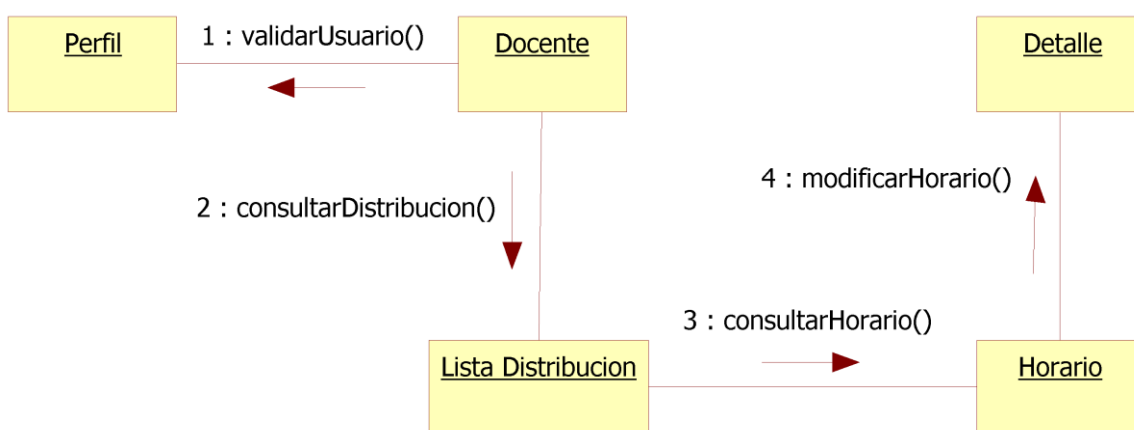


Figura 4.84. Modelo de Servicio – Modificar Horario.

4.2.2.7.4. MODULO DE INGRESO AL SISTEMA

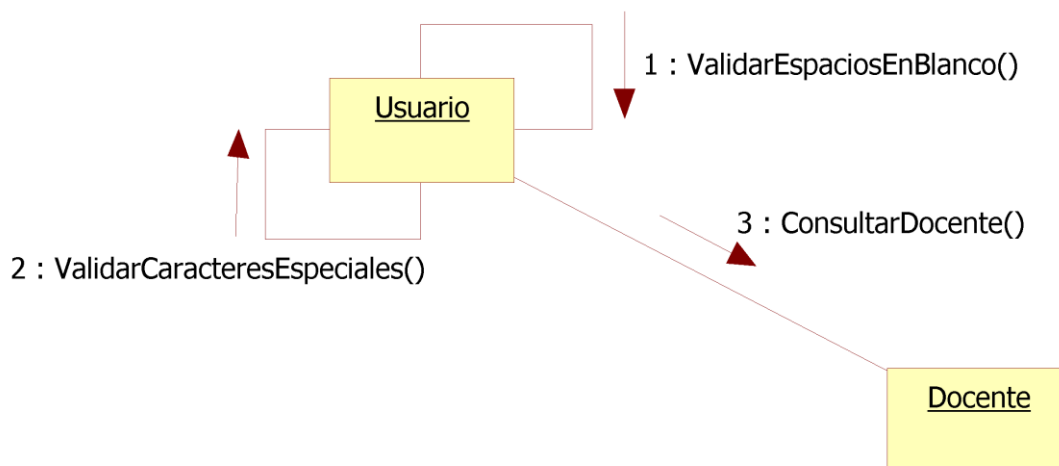


Figura 4.85. Modelo de Servicio – Consultar Docente.

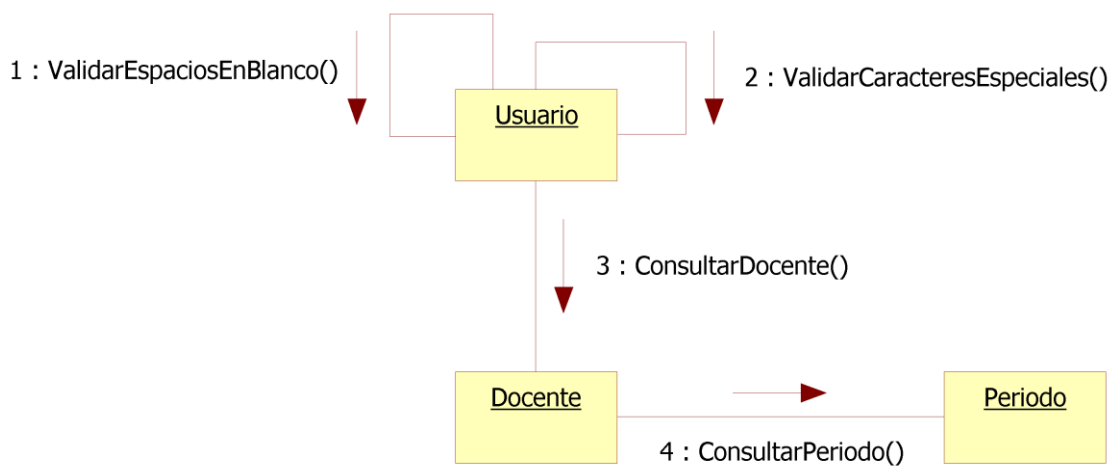


Figura 4.86. Modelo de Servicio – Consultar Periodo.

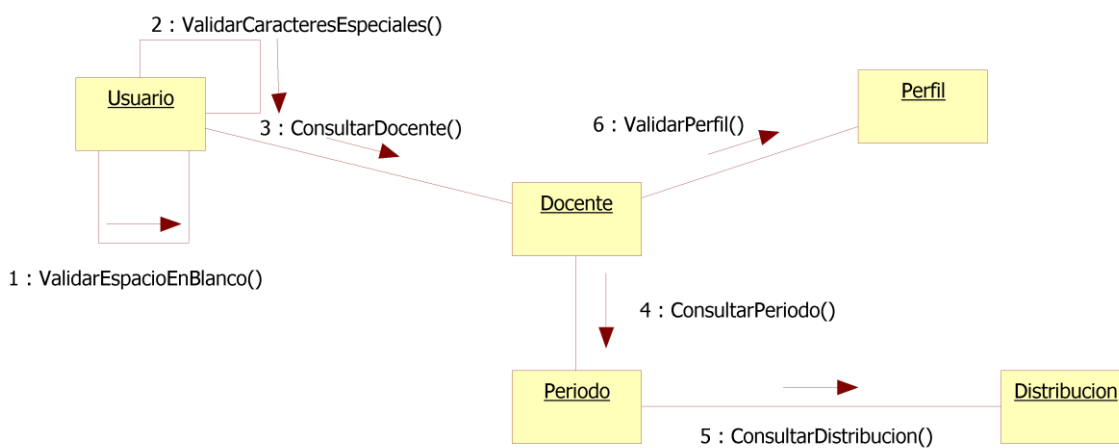


Figura 4.87. Modelo de Servicio – Iniciar Sesión

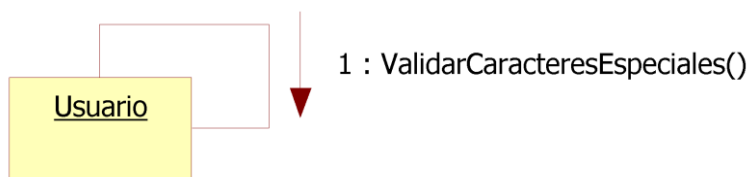


Figura 4.88. Modelo de Servicio – Validar Caracteres Especiales.

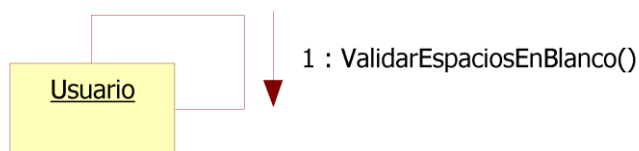


Figura 4.89. Modelo de Servicio – Validar Caracteres Especiales.

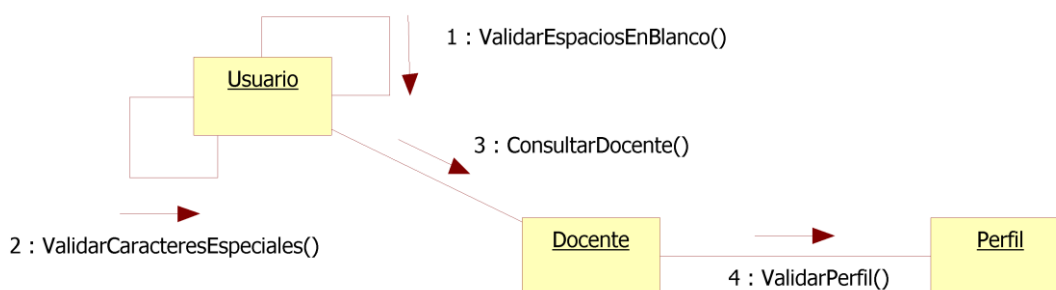


Figura 4.90. Modelo de Servicio – Validar Perfil.

4.2.2.7.5. MODULO DE JUSTIFICAR ASISTENCIA

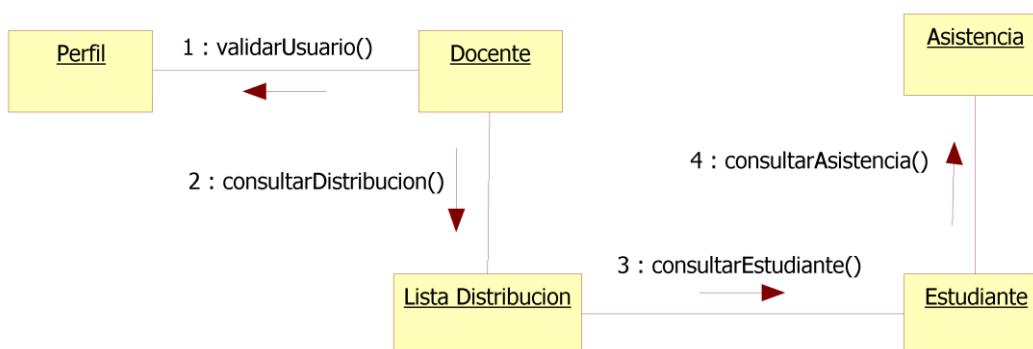


Figura 4.91. Modelo de Servicio – Consultar Asistencia.

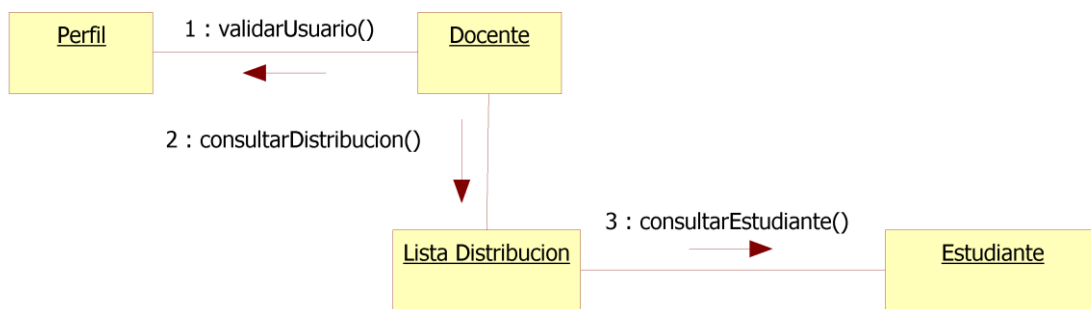


Figura 4.92. Modelo de Servicio – Consultar Estudiante.

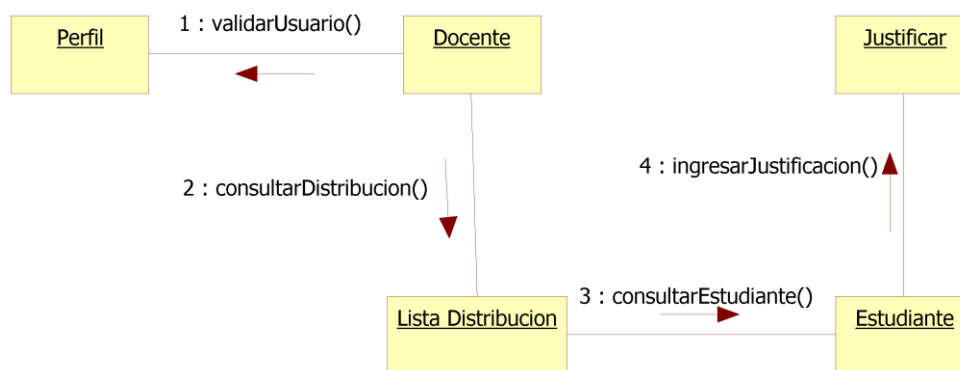


Figura 4.93. Modelo de Servicio – Ingresar Justificación.

4.2.2.7.6. MODULO DE REPORTE

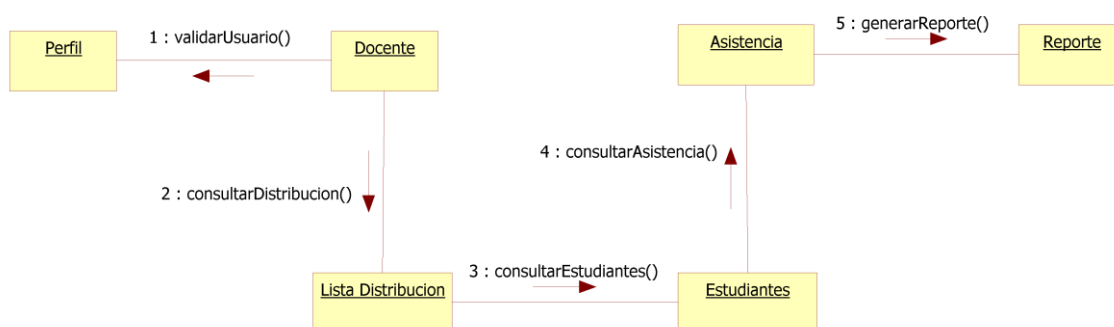


Figura 4.94. Modelo de Servicio – Ingresar Justificación.

4.2.2.7.7. MODULO DE BITÁCORA

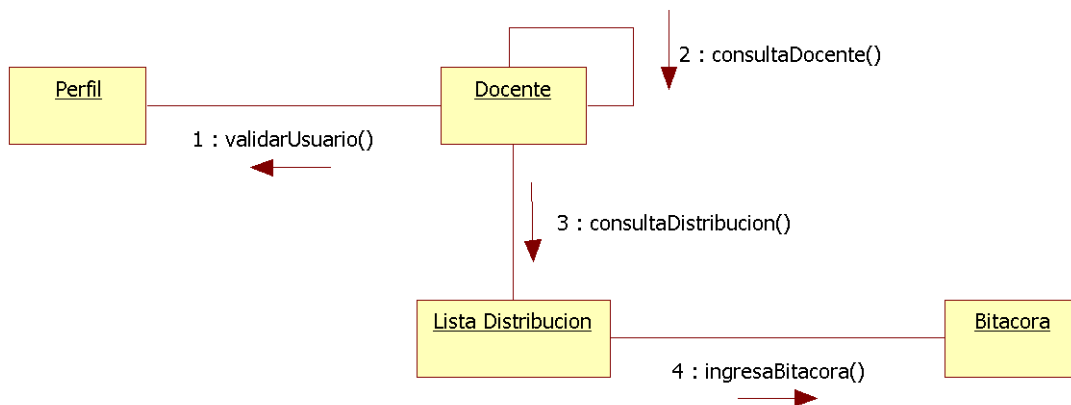


Figura 4.95. Modelo de Servicio – Ingresar Bitácora.

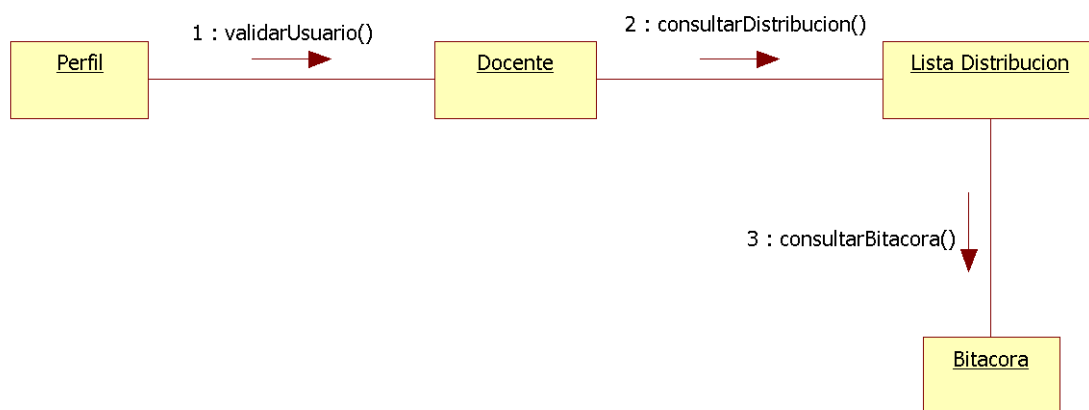


Figura 4.96. Modelo de Servicio – Consulta Bitácora.

4.2.2.7.8. MODULO DE TIEMPO

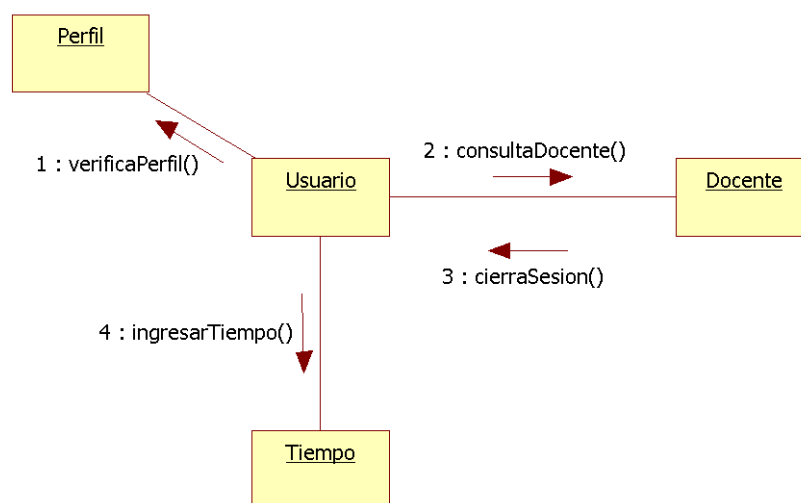


Figura 4.97. . Modelo de Servicio – Ingresar Tiempo

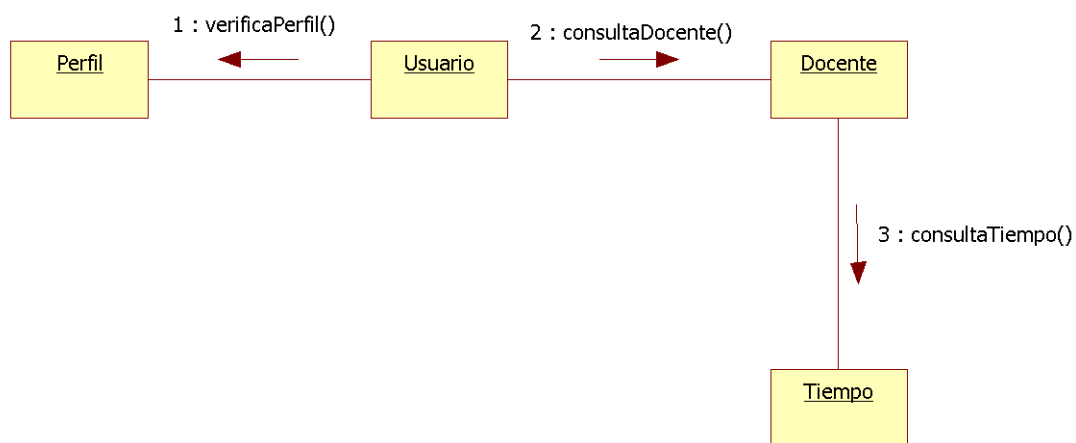


Figura 4.98. Modelo de Servicio – Consultar Tiempo

4.2.2.7.9. MODULO DE COORDINADOR

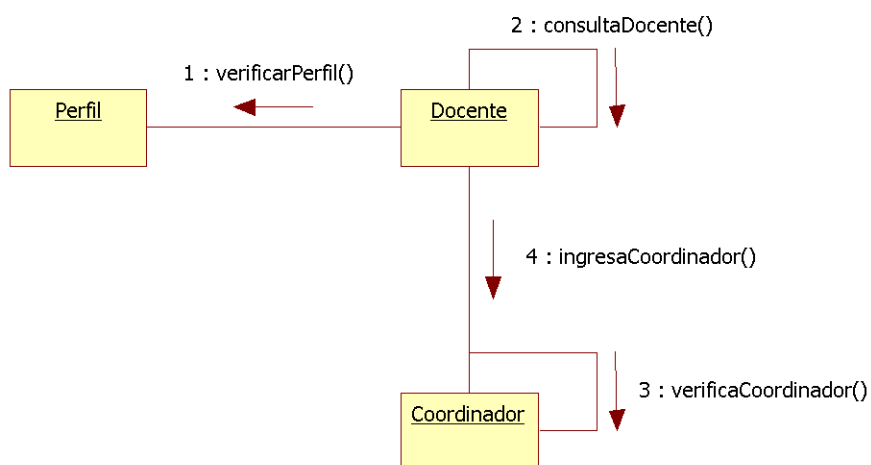


Figura 4.99. Modelo de Servicio – Ingresar Coordinador

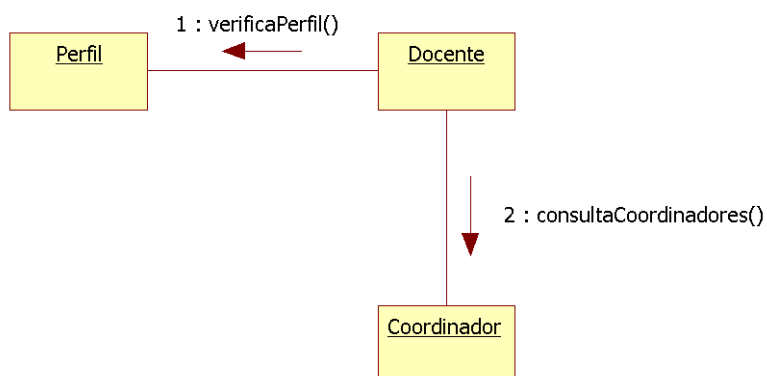


Figura 4.100. Modelo de Servicio – Consultar Coordinador.

4.2.2.7.10. MODULO DE PERSONALIZACIÓN.

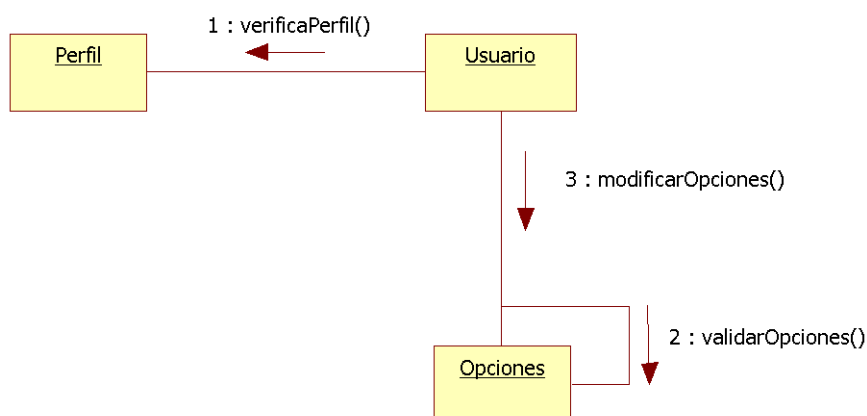


Figura 4.101. Modelo de Servicio – Modificar Opciones.

4.2.3. MODELO ESPECÍFICO DE LA PLATAFORMA

Los resultados que mostro esta iteración, se relacionaron directamente con las tecnologías que se implementaron en la aplicación web.

4.2.3.1. MODELO OBJETO RELACIONAL

El Modelo Objeto-Relacional definió la estructura, relaciones y tipos de datos de las unidades de información de la base de datos.

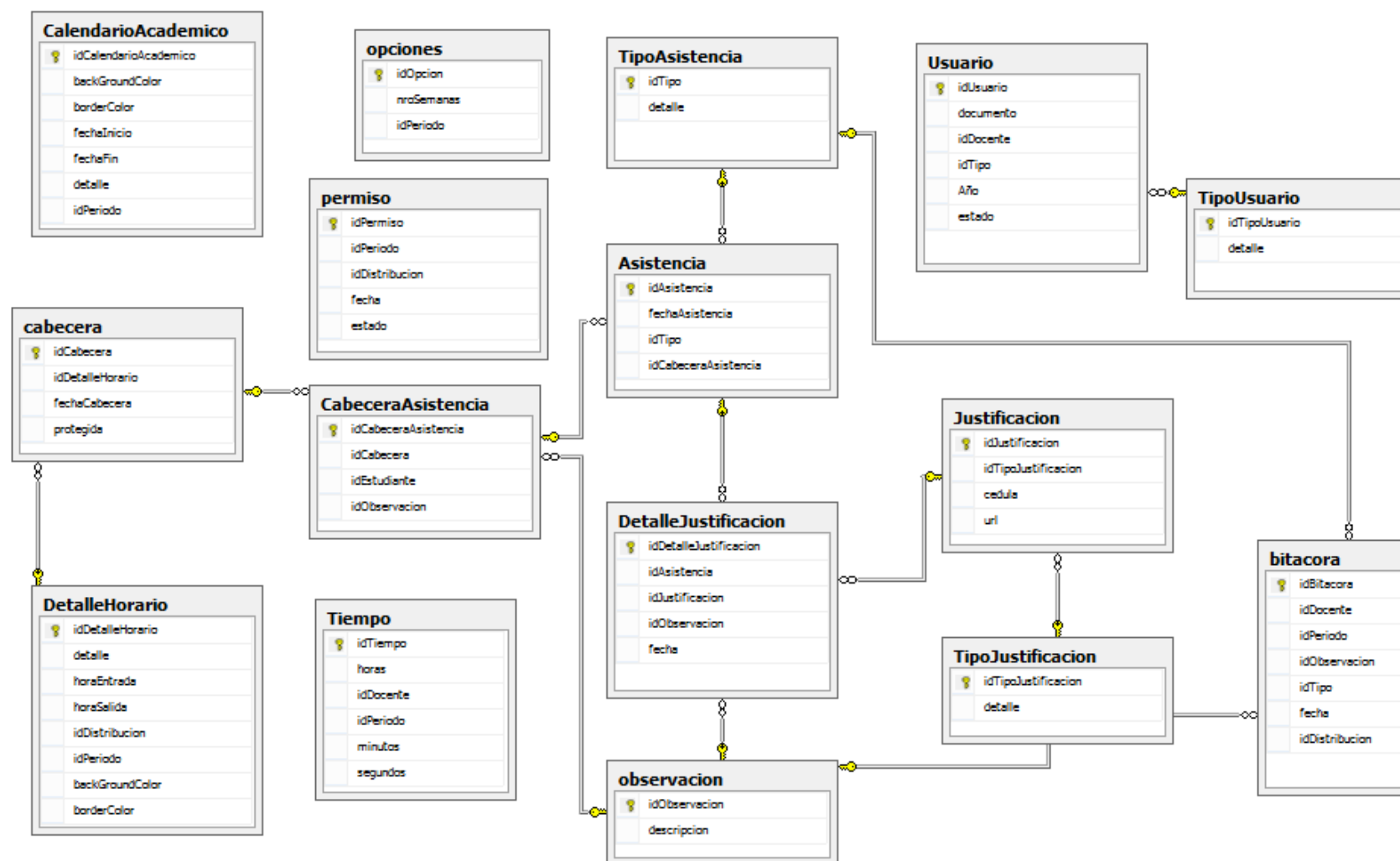


Figura 4.102. Modelo Objeto-Relacional.

4.2.3.2. MODELOS XML SCHEMAS.

Los fragmentos funcionales y estructurados están contruidos en este modelo para describir las restricciones de instancias, estructura interna de cada una de los componentes conceptuales del sistema (Anexo 2).

```
<xs:element name="consultarCabeceraAsistencia" type="consultarCabeceraAsistencia"/>
<xs:complexType name="consultarCabeceraAsistencia">
  <xs:sequence>
    <xs:element name="idCabeceraAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

Figura 4.103. Ejemplo de Código XML Schema para el fragmento consultarCabeceraAsistencia.

4.2.3.3. MODELO WSDL

La estructura de los servicios Web está planteada en un documento XML, que tiene seis partes, que son cabecera <definitions>, tipos de datos <types>, mensajes <message>, tipos de puerto <portType>, vínculos <binding> y los servicios <service>, definen el alcance del servicio web en la implementación (Anexo 3).

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="serviceCASYS"
  targetNamespace="http://www.exampleURI.com/casys"
  xmlns:tns="http://www.exampleURI.com/casys"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
```

Figura 4.104. Cabecera WSDL.

```
<wsdl:types>
  <xs:schema targetNamespace="http://www.exampleURI.com/casys"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:s1="http://www.exampleURI.com/casys">
    <xs:element name="consultarAsistencia" type="s1:consultarAsistencia"/>
    <xs:complexType name="consultarAsistencia">
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="id" type="xs:int" minOccurs="1" maxOccurs="1"/>
        <xs:element name="idAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="consultarAsistenciaResponse" type="s1:consultarAsistenciaResponse"/>
    <xs:complexType name="consultarAsistenciaResponse">
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
</wsdl:types>
```

Figura 4.105. Ejemplo de Código XML para los tipos de datos de los servicios.

```

<wsdl:message name="consultarAsistenciaSoapIn">
  <wsdl:part name="parameters" type="consultarAsistenciaResponse"/>
</wsdl:message>
<wsdl:message name="consultarAsistenciaSoapOut">
  <wsdl:part name="parameters" type="consultarAsistenciaResponse"/>
</wsdl:message>

```

Figura 4.106. Ejemplo de Código XML para los mensajes de las funciones.

```

<wsdl:portType name="asistenciaSOAP">
  <wsdl:operation name="consultarAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:consultarAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarAsistenciaSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ingresarAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:ingresarAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarAsistenciaSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="modificarAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:modificarAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:modificarAsistenciaSoapOut"/>
  </wsdl:operation>
</wsdl:portType>

```

Figura 4.107. Ejemplo de Código XML para las operaciones permitidas en los servicios.

```

<wsdl:binding name="asistenciaBindingSOAP" type="tns:asistenciaSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="modificarAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

Figura 4.108. Ejemplo de Código XML para especificación de protocolos de comunicación.

```

<wsdl:service name="asistenciaService">
  <wsdl:port name="asistenciaPortSOAP" binding="tns:asistenciaBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/asistencia.asmx"/>
  </wsdl:port>
</wsdl:service>

```

Figura 4.109. Ejemplo de Código XML correspondiente a los servicios Web.

4.3. DESARROLLO DEL SISTEMA

En el desarrollo del sistema se usaron dos herramientas de desarrollo el IDE Microsoft Visual Studio 2012 para codificar el sistema mediante el paradigma de desarrollo de programación por capas, en las que para este proyecto se representó mediante los contenedores del IDE que fueron dos: soluciones y proyectos; además para la base de datos se usó Microsoft SQL 2008 R2, para implementar el esquema tablas con sus relaciones, utilizando la información documentada de la metodología MIDAS.

4.3.1. CAPA DE DATOS

En la capa de datos se representó con un proyecto de bibliotecas de clases en la que contenía, un modelo de Entity Framework que es una tecnología de acceso a datos para poder obtener los procedimientos almacenados (stored procedure) y lograr la manipulación de los datos, representando así la conexión con la base de datos desarrollada para el propósito de CASYS, que se implementó previamente en el sistema manejador de base de datos Microsoft SQL Server, con el modelo de objeto relacional de la metodología MIDAS.

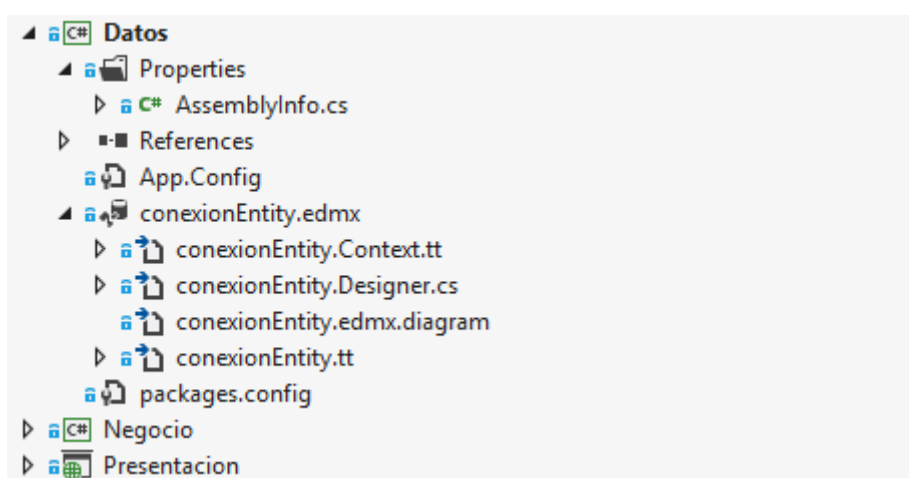


Figura 4.110. Estructura de la capa de Datos.

4.3.2. CAPA DE NEGOCIO

En esta capa se representó con un proyecto de bibliotecas de clases en cual se realizaron un total de 39 clases divididas en 2 partes: clases constructoras que abarcaron un total de 21 **¡Error! No se encuentra el origen de la referencia.** y su vez 18 objetos catálogos con los métodos para permitir la comunicación entre la capa de presentación y la capa de datos permitiendo realizar acciones como inserciones, consultas, modificaciones, eliminaciones y búsquedas, en primera instancia esta capa recibe una petición por parte de la capa de presentación (interfaces del sistema) ya sea cualquier acción, se accede al método del objeto asociado, el cual está relacionado con el stored procedure para poder manipular los datos, mediante la conexión de la capa datos.

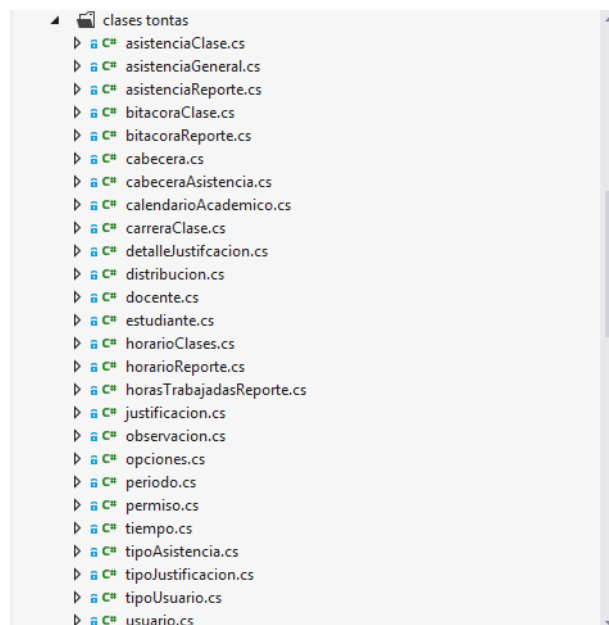


Figura 4.111. Objeto de Clases del sistema.

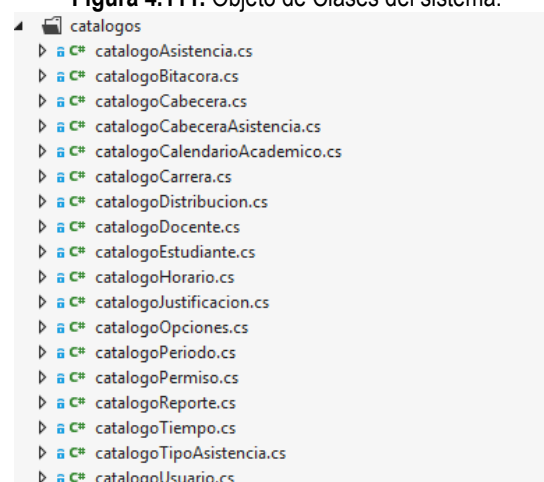


Figura 4.112. Objetos de Catálogos que se usaron en el sistema.

4.3.3. CAPA DE PRESENTACIÓN

En esta capa se personifico mediante un proyecto de aplicación Web vacía en la que se realizó el diseño de las interfaces del sistema de acuerdo a la Especificación de Requerimientos de Sistema, para lo cual se utilizó la plantilla institucional para que tuviera concordancia en el estilo con el sistema institucional, se utilizó Bootstrap como único framework de diseño, además de varias librerías complementarias de tipo JS y JQuery para poder dinamizar el sistema. Para la realización de las mismas se tomó en cuenta los casos de usabilidad para una buena presentación del sistema y que el mismo sea fácil de manipular.



Figura 4.113. Formulario de bienvenida al sistema.

4.4. VERIFICACIÓN DEL SISTEMA

La validación del sistema se realizó en dos etapas: la autoevaluación del sistema por parte de los autores, y la comprobación del aplicativo por parte de un grupo de docentes, en la primera se ejecutaron diferentes pruebas con datos ficticios a los diferentes módulos con los que cuenta el sistema, logrando comprobar errores de tipo lógico, para a su vez corregirlos, y pasar a la segunda etapa, empezando por la capacitación a los docentes seleccionados para realizar la tarea de utilizar el sistema, y poder observar errores e incoherencia al momento de tratar la información real, en dicha capacitación se entregó un Manual de Usuario del Aplicativo (Anexo 4), que mostraba

Reportes de tiempo

Inicio / Reportes de tiempo de uso del sistema

Reporte

30 DE MARZO - 27 DE AGOSTO 2015 (INICIO DE CLASES LUNES 13 DE ABRIL) INFORMÁTICA

Generar Reporte

ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ
CARRERA DE INFORMÁTICA
PERIODO 30 DE MARZO - 27 DE AGOSTO 2015 (INICIO DE CLASES LUNES 13 DE ABRIL)
REGISTRO DE TIEMPO DE USO DE CASYS

DOCENTES	HORAS	MINUTOS	SEGUNDOS	
VICTOR JOEL PINARGOTE BRAVO		0	92	29
JESSICA JOHANNA MORALES CARRILLO	2	35	60	
ANGEL ALBERTO VELEZ MERO	0	24	32	

22/07/2015
9:29:07

1

Figura 4.115. Reporte de uso del sistema por parte de los docentes.

Para corroborar esta comprobación se comparó el tiempo que lleva entrega de reportes de forma manualmente, y el tiempo que lleva usando el sistema; en el sistema manual los docentes tenían que presentar un documento impreso lo cual por cuestiones de ocupaciones de los mismo en muchos caso, completar este proceso demoraba varios días, con el uso del sistema CASYS los tiempos para poder acceder a los reportes de asistencias mejoraron notablemente debido a que cuando inmediatamente se termina el periodo de clases normales, los reportes podrán ser generados por cada uno de los usuarios que tengan acceso a ellos cuando lo deseen.

Cuadro 4.3. Comparación de Tiempo entre los procesos.

	Manual	Sistema CASYS
Entrega de reportes.	24 horas o más.	3 minutos máximos.
Tomar asistencia.	5 minutos	3 minutos.

De acuerdo a las pruebas realizadas se obtuvo un promedio del tiempo de respuesta y se constató que efectivamente el tiempo de respuesta por parte del

sistema beneficiara en un 99,80%, sin duda alguna tanto en los aspectos de toma de asistencias como al acceso a los reportes, ayudando a la toma de decisiones.

Todo desarrollo de sistema se debe realizar de manera organizada para alcanzar los requerimientos y la solución del problema, por lo que según Castro (2007) aplicar un estudio de las principales metodologías para el desarrollo de Sistemas de información web, propone una especiación de las funcionalidades y aspectos de manera general utilizando UML.

Para realizar la adaptación en el paradigma de desarrollo escogido, en este trabajo se aplicó la metodología de desarrollo ágil MIDAS, en que propone modelos y esquemas para describir cada una de las partes del sistema, logrando describir cada característica del sistema, para luego codificar estas funcionalidades de manera ordenada, dado que Puertas y Robayo (2007) plantea que el uso de un sistema informático provee herramientas para la gestión de procesos, con capacidad de obtener estadísticas, entregando información de suma importancia, beneficiando a los usuarios en la actividad automatizada. Además se recomiendan utilizar estándares para los nombres de variables, controles y objetos, para ayudar a estructurar un código limpio y de fácil depuración, por lo que se utilizó el estilo de escritura lower Camel Case, de acuerdo a estructura planteada en la Especificación de Requerimientos de Software, logrando cumplir aspectos recopilados en los inicios de este trabajo.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La utilización de un estándar de Especificación de Requerimientos de Software, permite que la información analizada tenga un formato que facilite la explicación de las necesidades y requisitos de la solución tecnológica.
- El empleo de una metodología de desarrollo de sistemas de información, permite conceptualizar los diferentes procesos para obtener una aplicación web de calidad.
- Esquematizar el modelo independiente de computación incluyendo las características y objetos del sistema tomando en cuenta cada uno de los aspectos recolectados en la Especificación de Requerimientos de Software, determina el alcance que tendrán las funciones del módulo web.
- La definición del Modelo Independiente de la Plataforma mediante el modelado de los aspectos del sistema usando software UML, simplifica la documentación del desarrollo de una aplicación web.
- Estructurar los Modelos Específicos de Plataforma, ayuda en los diseños de los aspectos tecnológicos a implementar en el sistema.

5.2. RECOMENDACIONES

- Utilizar el formato aplicando un lenguaje informal, para que sea fácilmente comprensible por cualquier persona sin tener conocimientos técnicos, o esté involucrada en el desarrollo del sistema.
- Usar herramientas UML que tenga extensiones para otras tecnologías, ya que facilita el proceso de conceptualización del aplicativo web.
- Esquematizar las funciones generales que tendrán los sistemas, para permitir establecer límites en las etapas de desarrollo.
- La definición de los diferentes aspectos del desarrollo de software mediante herramientas UML, se debe de realizar de acuerdo a la visión de metodología que se está utilizando para el desarrollo.
- La estructuración de los detalles de las tecnologías a implementar en las plataformas del sistema, deben tener cada uno atributos e información que se haya realizado en la documentación.

BIBLIOGRAFÍA

- Alegsa, L. 2010a. Definición de módulo web. (En línea). Santa Fe-Argentina. Consultado, 21 de jun. 2014. Formato HTML. Disponible en: <http://www.alegsa.com.ar/Dic/modulo.php>
- _____. 2010b. Definición de servicio web. (En línea). Santa Fe-Argentina. Consultado, 21 de jul. 2014. Formato HTML. Disponible en: <http://www.alegsa.com.ar/Dic/servicio%20web.php#sthash.elKAfXg4.dpuf>
- _____, 2010c. Optimización de aplicaciones. (En línea). AR. Consultado, 16 de jun. 2014. Formato HTML. Disponible en <http://www.alegsa.com.ar/Dic/optimizar.php>
- Alvarez, M. 2001. Concepto de ASP. (En Línea). Consultado, 6 de may. 2014. Formato HTML. Disponible en <http://www.desarrolloweb.com/articulos/393.php>.
- Anaya, I. 2013. Investigación Documental. (En línea). San Francisco-USA. Consultado, 7 de dic. 2014. Formato HTML. Disponible en <http://es.slideshare.net/FerGiza/investigacin-documental-16405056>.
- Asamblea Constituyente, 2008. Constitución de la República del Ecuador. (En línea). EC. Consultado, 16 de jun. 2014. Formato PDF. Disponible en http://www.asambleanacional.gov.ec/documentos/constitucion_de_bolsillo.pdf
- Armas, M. 2009. Model Driven Architecture (MDA) (En línea). México. Consultado, 21 de jun. 2014. Formato HTML. Disponible en <http://sg.com.mx/content/view/861>
- Barragán, R. 2005. El portafolio, metodología de evaluación y aprendizaje de cara al nuevo Espacio Europeo de Educación Superior. Una experiencia práctica en la Universidad de Sevilla. España. Revista Latinoamericana de Tecnología Educativa. Vol 4. Núm. 1. p. 121-129.
- Bassi, S. 2013. Python en 8 clases: Aprendiendo a programar con Python. JSON. Genes Digitales. Cuarta edición. Argentina. P. 38
- Beck, K. 1999. Embracing Change with eXtreme Programming. USA. Computer, Vol 32. Núm. 10. p. 70-77.

- Camps, R; Casillas, L; Costal, D; Gibert, M; Martín, C; Pérez, O. 2005. Introducción a Base de datos. España. Eureka Media. p. 5.
- Caivano, R y Villoria, L. 2009. Aplicaciones web 2.0 Google docs. Lenguajes de programación. Argentina. Eduvim. p. 18-20.
- Castro, M. 2007. Aproximación MDA para el desarrollo orientado a servicios de sistemas de información web: del modelo de negocio al modelo de composición de servicios web. Tesis. Doctorado. Universidad Rey Juan Carlos. Móstoles-Madrid. ES. p. 97-117.
- CCIA (Ciencias de la Computación e Inteligencia Artificial). 2008. Definición de servicio. (En línea). Sevilla-España. Consultado, 21 de jul. 2014. Formato PDF. Disponible en <http://ccia.ei.uvigo.es/docencia/SCS/0910/transparencias/Tema4.pdf>
- Cobo, A. 2008. Diseño y programación de bases de datos. Concepto de base de datos. España. Visión libro. p. 7.
- Duarte, A. 2007. Mateurística. Concepto de Optimización. DYKISON S.L. Madrid-España. DYKISON S.L. p. 1.
- E-lemental. 2010. Concepto aplicación web. (En línea). EC. Consultado, 16 de jun. 2014. Formato HTML. Disponible en <http://www.e-lemental.com.pe/que-es-una-aplicacion-web/>
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López), s.f. Reglamento Integral del Desempeño del Personal Académico de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López. 1 ed. Calceta-Manabí, EC.
- Prendes, M y Sánchez, M. 2008. Portafolio electrónico. Universidad de Sevilla, España. Red de Revistas Científicas de América Latina. Revista de Medios y Educación. Núm. 32. p. 21-34.
- Esquivá, A. 2013. JSON, ¿Qué es y para qué sirve? (En línea). Consultado, 23 de jul. 2014. Formato HTML. Disponible en <http://geekytheory.com/json-i-que-es-y-para-que-sirve-json/>
- Fons, M. 2006. Leer y escribir para vivir. Barcelona-España. Graó, de Irif, S.L y la galera. p. 112.

- Fontela, A. 2013. BOOTSTRAP. (En línea). Consultado, 7 de jul. 2014. Formato HTML. Disponible en <http://openwebcms.es/2013/que-es-bootstrap/>
- Galván, V y Yesith, J. 2013. Modelo de sistema de información para apoyar la gestión de proyectos de investigación en grupos de investigación. Pereira, CO. Red de Revistas Científicas de América Latina. Vol 18. Núm. 4. p. 690-697.
- García, F. 2005. El papel de los portafolios electrónicos en la enseñanza aprendizaje de las lenguas. Murcia, ES. Glosas Didácticas. Núm. 14. p. 113-119.
- Gómez, A y Martínez, N. 1997. Los sistemas de información en la empresa. Conceptos de Bases de Datos. España. Servicio de publicaciones de la ciudad de Oviedo. p. 80.
- Gonzales, M. 2008. Estándar IEEE 830. (En línea). Argentina. Consultado, 7 de dic. 2014. Formato HTML. Disponible en <http://ingsoftudb.blogspot.com/>
- González, L. 2006. 10 pasó para aumentar la rentabilidad. Qué es un proceso. Díaz de Santos S.A. Argentina. Díaz de Santos S.A. p. 17.
- González, R; Canovas, R; Infante, A; Ortiz, R. 2013. Siscorfi: Una aplicación web para el control de los recursos filogenéticos. Colombia. Red de Revistas Científicas de América Latina. Vol 34. Núm. 3. p. 25-31.
- Gutiérrez, J. 2012. BOOTSTRAP. (En línea). Consultado, 7 de jul. 2014. Formato HTML. Disponible en <http://www.genbetadev.com/frameworks/bootstrap>.
- Hilda, E. 1996. El portafolio como estrategia para la evaluación. Teoría y didáctica de la lengua y de la literatura. Puerto Rico. Universidad Interamericana de Puerto Rico. Núm. 8. p. 89- 96.
- IEEE (Institute of Electrical and Electronics Engineers), 2009. IEEE Recommended Practice for Software Requirements Specifications Standard IEEE-830-1998. New York, USA.
- Junta de Castilla y León (2006). Trabajando con los Procesos: Guía para la gestión por procesos. Valladolid: España

Luján, S, 2002. Programación de aplicaciones web: historia, principios básicos y clientes web. España. Editorial Club Universitario. p. 7-9.

Microsoft. 2007a. Introducción a Visual Studio. (En Línea). Consultado, 6 de may. 2014. Formato HTML. Disponible en [http://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.90).aspx).

_____ 2007b. Concepto ASP.NET. (En Línea). Consultado el 6 de Mayo del 2014. Formato HTML. Disponible <http://www.microsoft.com/web/platform/framework.asp>

Montilva, J; Rojas, M; Orjuela, A. 2011. RDOA-WS: repositorio distribuido de objetos de aprendizaje soportado con servicios web. Colombia. Revista Avances en Sistemas e Informática. Vol. 8. Núm. 2. pp. 183-189.

Navarro, F; Corrales, J; Villar, M; Sampalo De La Torre, A; Cifredo, E; Olivas, L. 2006. Técnicos de soporte informático. Qué es ASP. España. MAD. S.L. p. 363.

Overmyer, S. 2000. What's Different about Requirements Engineering for Web Sites? Inglaterra. Requirements Engineering. Vol. 5. p. 62-65

Pérez, C, y Juan F. 2006. Control de la gestión empresarial. 6 ed. España. Libros Profesionales de Empresa/ ESIC Editorial. p. 21.

Pérez, V. 2007. Concepto de base de datos. (En línea). Cuba. Consultado, 27 de abr. 2014. Formato HTML. Disponible en <http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/>.

Ponce, M. 2010? Control de Gestión. (En línea). San Isidro-Lima. Consultado, 16 de jun. 2014. Formato PDF. Disponible en <http://www.partnerconsulting.com.pe/UserFiles/File/Articulos/Que%20es%20un%20sistema%20de%20gestion%20de%20control.pdf>

Poveda, C; Díaz, G; Abendaño, A; Benalcazar, A; Araujo de Solís, Susana; Arboleda, R. 1994. Informe de la Organización de Estados Americanos- Ministerio de Educación del Ecuador. (En línea). Ecuador. Consultado, 21 de junio 2014. Formato PDF. <http://www.oei.es/quipu/ecuador/#sis>

Puertas, V. y Robayo, F. 2007. Estudio de la metodología MIDAS y desarrollo del sistema de administración de competencias de basquetbol (BASYS) para la facultad de educación Física, deportes y recreación de la ESPE.

Ingeniera en sistemas e informática Doctorado. ESPE. Sangolquí-Quito. EC. p. 32-85.

Ramos, A y Ramos, J. 2011. Aplicaciones web. España. Quality SGE. p. 2-9.

Rocha, D. 2014. Electrónica e informática aplicada. Que es un módulo. Alsina. p.191.

Silberschatz, A; Korth, H; Sudarshan, S. 2006. Fundamentos de Bases de Datos. Fernando Sáenz Pérez. McGraw-Hill. Madrid-España. Concepción Fernández Madrid. p. 1.

Sommerville, I. 2001. Ingeniería de Software. 6 Ed. España. Pearson Educación. p. 60-79.

Sudha, N. y Sujatha, S. 2011. Integrating SOAP and Web Services. Simple Object Access Protocol (SOAP). River Publishers. Dinamarca. p. 121-122.

Turk, D; France, R; Rumpe, B; Georg, G; 2002. Model-Driven Approach to Software Development. Advances in Object-Oriented Information Systems. Francis. Springer-Verlag. p. 229-230.

Van, C. 2005. ¿Qué es la web 2.0?. (En línea). USA. Consultado, 27 de abr. 2014. Formato HTML. Disponible en <http://www.maestrosdelweb.com/editorial/web2/>.

Vela, B. 2003. MIDAS/DB: Una Metodología basada en Modelos para el Desarrollo de la Dimensión Estructural de Sistemas de Información Web. Tesis Doctoral. URJC. Madrid. ES. p 41-119.

Zorrilla, M y Vela, B. 2007. Una aproximación dirigida por modelos para diseñar y construir esquemas XML. Un caso de estudio. Colombia. Red de Revistas Científicas de América Latina. Vol 4. Pág.3.

ANEXOS

ANEXO 1

Entrevista Informal

Objetivo: Obtener información relevante para el desarrollo de la tesis de grado módulo web para el control de asistencia de los estudiantes de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López

Dirigida: Ing. Ángel Alberto Vélez Mero, Mgs

1. ¿Cuáles son los principales problemas de automatización de procesos en el departamento?

En la actualidad, para mejorar los procesos de acreditación, se necesitan automatizar los sistemas que se llevan realizando como los planes de clases, políticas el curso, registro de asistencia, registro de evaluaciones, bancos de reactivos, informe del docente, informe de coherencia y reporte estudiantil del seguimiento, todos son indicadores para la acreditación y evaluación, y se necesita un mejor manejo de la información que se trata en estos indicadores.

2. De los problemas presentados: ¿Cuáles son los que se necesitan con mayor urgencia ser automatizados?

Cada uno de los problemas que se mencionó, se necesita que estén automatizados para un optimizar el análisis de la información que muestran los mismos. Se requiere solucionar cada uno de estos problemas por etapas, en la actualidad se proponen para los estudiantes de la carrera de informática el registro de asistencia, registro de evaluaciones, como problemas prioritarios.

3. ¿Cuáles son las actividades que llevan los profesores para entregar los registros de asistencia?

Los profesores deberían entregar en teoría según el Reglamento Integral del Desempeño del Personal Académico, el reporte de asistencia de manera mensual, sin embargo, ellos deben de tomar la asistencia diariamente y al final de cada mes imprimir este reporte para poder entregarlo.

4. ¿Se podrán modificar las asistencias después de las 48 horas reglamentarias en el manual de procesos?

Lo que se quiere que haga el sistema es precisamente controlar las irregularidades que se presentan con respecto a tomar las asistencia, al justificar las faltas, al presentar los reportes y que se cumplan con las disposiciones impuestas en el manual de procesos de la universidad así que no se deberían de modificar las asistencias después de haber cumplido con el tiempo reglamentario.

5. ¿El calendario académico será una herramienta para determinar días no laborables y así automatizar el modulo web?

Efectivamente el calendario académico debería de ser una herramienta que determine cuáles son los días que no son laborables para que la asistencia de los estudiantes en esos días no se vea afectada.

6. Para crear un correcto funcionamiento del sistema. ¿Cuál es la sugerencia que podría realizar a los autores, dado el cambiante horario de clases en los primeros meses del semestre?

Ingresar los diferentes horarios de clases, de las distintas cátedras de los docentes, y que solo se tome en cuenta el último horario ingresado, pero eso como media de solución, actualmente en la universidad está trabajando para evitar modificar los horarios en las diferentes carreras.

7. ¿Cuáles serán los datos necesarios para que se almacenen en el módulo?

La mayoría de los datos relevantes se encuentran en el formato de la hoja de cálculo que se utiliza en la actualidad.

8. De los datos ingresados. ¿Cuáles son los reportes que se deben de presentar?

Como pueden ver en la hoja de cálculo los datos estadísticos son los necesarios para un análisis, se necesita también la estadística de la asistencia de los estudiantes de manera general.

9. ¿Cómo se deben de presentar estos reportes?

Estos reportes deben de ser presentados en forma de pasteles estadísticos de manera porcentual para su mejor entendimiento.

8. ¿Tiene algún tipo de preferencia de diseño del aplicativo web?

Pueden utilizar el mismo diseño que el aplicativo de la universidad está utilizando para que sea familiar para el usuario. Pero tomando en cuenta que debe ser funcional, que se tomen en consideración en datos del docente, el semestre y la materia.

ANEXO 2

MODELOS XML SCHEMAS

```

<xs:element name="verificarPerfil" type="verificarPerfil"/>
<xs:complexType name="verificarPerfil">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 116. Código XML Schema para el fragmento verificarPerfil.

```

<xs:element name="consultarDocente" type="consultarDocente"/>
<xs:complexType name="consultarDocente">
  <xs:sequence>
    <xs:element name="idDocente" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 117. Código XML Schema para el fragmento consultarDocente

```

<xs:element name="consultarTipoJustificacion" type="consultarTipoJustificacion"/>
<xs:complexType name="consultarTipoJustificacion">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 118. Código XML Schema para el fragmento consultarTipoJustificacion

```

<xs:element name="consultarDistribucion" type="consultarDistribucion"/>
<xs:complexType name="consultarDistribucion">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 119. Código XML Schema para el fragmento consultarDistribucion

```

<xs:element name="consultarPeriodo" type="consultarPeriodo"/>
<xs:complexType name="consultarPeriodo">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 120. Código XML Schema para el fragmento consultarPeriodo

```

<xs:element name="consultarHorario" type="consultarHorario"/>
<xs:complexType name="consultarHorario">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleHorario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 121. Código XML Schema para el fragmento consultarHorario

```

<xs:element name="consultarCaberaAsistencia" type="consultarCaberaAsistencia"/>
<xs:complexType name="consultarCaberaAsistencia">
  <xs:sequence>
    <xs:element name="idCabeceraAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 122. Código XML Schema para el fragmento consultarCabeceraAsistencia

```

<xs:element name="consultarJustificacion" type="consultarJustificacion"/>
<xs:complexType name="consultarJustificacion">
  <xs:sequence>
    <xs:element name="idJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="url" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 123. Código XML Schema para el fragmento consultarJustificacion

```

<xs:element name="consultarDetalleJustificacion" type="consultarDetalleJustificacion"/>
<xs:complexType name="consultarDetalleJustificacion">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 124. Código XML Schema para el fragmento consultarDetalleJustificacion

```

<xs:element name="consultarAsistencia" type="consultarAsistencia"/>
<xs:complexType name="consultarAsistencia">
  <xs:sequence>
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 125. Código XML Schema para el fragmento consultarAsistencia

```

<xs:element name="consultarEstudiante" type="consultarEstudiante"/>
<xs:complexType name="consultarEstudiante">
  <xs:sequence>
    <xs:element name="apellido" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="nombre" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 126. Código XML Schema para el fragmento consultarEstudiante

```

<xs:element name="consultarTipoAsistencia" type="consultarTipoAsistencia"/>
<xs:complexType name="consultarTipoAsistencia">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 127. Código XML Schema para el fragmento consultarTipoAsistencia

```

<xs:element name="consultarEventos" type="consultarEventos"/>
<xs:complexType name="consultarEventos">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCalendarioAcademico" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 128. Código XML Schema para el fragmento consultarEventos

```

<xs:element name="ingresarJustificacion" type="ingresarJustificacion"/>
<xs:complexType name="ingresarJustificacion">
  <xs:sequence>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idUserario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="url" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 129. Código XML Schema para el fragmento ingresarJustificacion

```

<xs:element name="modificarAsistencia" type="modificarAsistencia"/>
<xs:complexType name="modificarAsistencia">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 130. Código XML Schema para el fragmento modificarAsistencia

```

<xs:element name="ingresarAsistencia" type="ingresarAsistencia"/>
<xs:complexType name="ingresarAsistencia">
  <xs:sequence>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 131. Código XML Schema para el fragmento ingresarAsistencia

```

<xs:element name="ingresarHorario" type="ingresarHorario"/>
<xs:complexType name="ingresarHorario">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 132. Código XML Schema para el fragmento ingresarHorario

```

<xs:element name="modificarHorario" type="modificarHorario"/>
<xs:complexType name="modificarHorario">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleHorario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 133. Código XML Schema para el fragmento modificarHorario

```

<xs:complexType name="ingresarEvento">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEvento" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 134. Código XML Schema para el fragmento ingresarEvento

```

<xs:element name="modificarEvento" type="modificarEvento"/>
<xs:complexType name="modificarEvento">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEvento" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 135. Código XML Schema para el fragmento modificarEvento

```

<xs:element name="iniciarSesion" type="iniciarSesion"/>
<xs:complexType name="iniciarSesion">
  <xs:sequence>
    <xs:element name="contraseña" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="usuario" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 136. Código XML Schema para el fragmento iniciarSesion

```

<xs:element name="consultarCoordinador" type="s1:consultarCoordinador"/>
<xs:complexType name="consultarCoordinador">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="anio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="documento" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estado" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCoordinador" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 137. Código XML Schema para el fragmento consultarCoordinador

```

<xs:element name="consultarTiempo" type="s1:consultarTiempo"/>
<xs:complexType name="consultarTiempo">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="horas" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCarrera" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTiempo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minutos" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="segundos" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 138. Código XML Schema para el fragmento consultarTiempo

```

<xs:element name="consultarBitacora" type="s1:consultarBitacora"/>
<xs:complexType name="consultarBitacora">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idBitacora" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idObservacion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 139. Código XML Schema para el fragmento consultarBitacora

```

<xs:element name="modificarPersonalizacion" type="s1:modificarPersonalizacion"/>
<xs:complexType name="modificarPersonalizacion">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 140. Código XML Schema para el fragmento modificarPersonalizacion

```

<xs:element name="ingresarTiempo" type="s1:ingresarTiempo"/>
<xs:complexType name="ingresarTiempo">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="horas" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCarrera" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minutos" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="segundos" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 141. Código XML Schema para el fragmento ingresarTiempo

```

<xs:element name="ingresarCoordinador" type="s1:ingresarCoordinador"/>
<xs:complexType name="ingresarCoordinador">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="anio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="documento" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estado" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 142. Código XML Schema para el fragmento ingresarCoordinador.

```

<xs:element name="ingresarBitacora" type="s1:ingresarBitacora"/>
<xs:complexType name="ingresarBitacora">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idObservacion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 143. Código XML Schema para el fragmento ingresarBitacora

ANEXO 3

MODELO WSDL


```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="serviceCASYS"
  targetNamespace="http://www.exampleURI.com/casys"
  xmlns:tns="http://www.exampleURI.com/casys"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

```

Figura 144. Cabecera WSDL

```

<wsdl:types>
  <xs:schema targetNamespace="http://www.exampleURI.com/casys"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:s1="http://www.exampleURI.com/casys">
    <xs:element name="consultarAsistencia" type="s1:consultarAsistencia"/>
    <xs:complexType name="consultarAsistencia">
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="id" type="xs:int" minOccurs="1" maxOccurs="1"/>
        <xs:element name="idAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="consultarAsistenciaResponse" type="s1:consultarAsistenciaResponse"/>
    <xs:complexType name="consultarAsistenciaResponse">
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="consultarCaberaAsistencia" type="s1:consultarCaberaAsistencia"/>
    <xs:complexType name="consultarCaberaAsistencia">
      <xs:sequence>
        <xs:element name="idCabeceraAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
        <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
        <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="consultarDetalleJustificacion" type="s1:consultarDetalleJustificacion"/>
    <xs:complexType name="consultarDetalleJustificacion">
      <xs:sequence>
        <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
        <xs:element name="idDetalleJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
        <xs:element name="idJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>

```

```

<xs:element name="consultarDistribucion" type="s1:consultarDistribucion"/>
<xs:complexType name="consultarDistribucion">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarEventoResponse" type="s1:modificarEventoResponse"/>
<xs:complexType name="modificarEventoResponse">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEvento" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarHorarioResponse" type="s1:modificarHorarioResponse"/>
<xs:complexType name="modificarHorarioResponse">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleHorario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="verificarPerfilResponse" type="s1:verificarPerfilResponse"/>
<xs:complexType name="verificarPerfilResponse">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarDocente" type="s1:consultarDocente"/>
<xs:complexType name="consultarDocente">
  <xs:sequence>
    <xs:element name="idDocente" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarEstudiante" type="s1:consultarEstudiante"/>
<xs:complexType name="consultarEstudiante">
  <xs:sequence>
    <xs:element name="apellido" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="nombre" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarEventos" type="s1:consultarEventos"/>
<xs:complexType name="consultarEventos">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCalendarioAcademico" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarHorario" type="s1:consultarHorario"/>
<xs:complexType name="consultarHorario">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleHorario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="consultarJustificacion" type="s1:consultarJustificacion"/>
<xs:complexType name="consultarJustificacion">
  <xs:sequence>
    <xs:element name="idJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="url" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarPeriodo" type="s1:consultarPeriodo"/>
<xs:complexType name="consultarPeriodo">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarTipoAsistencia" type="s1:consultarTipoAsistencia"/>
<xs:complexType name="consultarTipoAsistencia">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarTipoJustificacion" type="s1:consultarTipoJustificacion"/>
<xs:complexType name="consultarTipoJustificacion">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarAsistencia" type="s1:ingresarAsistencia"/>
<xs:complexType name="ingresarAsistencia">
  <xs:sequence>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarEvento" type="s1:ingresarEvento"/>
<xs:complexType name="ingresarEvento">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEvento" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarHorario" type="s1:ingresarHorario"/>
<xs:complexType name="ingresarHorario">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarJustificacion" type="s1:ingresarJustificacion"/>
<xs:complexType name="ingresarJustificacion">
  <xs:sequence>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="url" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="iniciarSesion" type="s1:iniciarSesion"/>
<xs:complexType name="iniciarSesion">
  <xs:sequence>
    <xs:element name="contraseña" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="usuario" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="modificarAsistencia" type="s1:modificarAsistencia"/>
<xs:complexType name="modificarAsistencia">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarEvento" type="s1:modificarEvento"/>
<xs:complexType name="modificarEvento">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEvento" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarHorario" type="s1:modificarHorario"/>
<xs:complexType name="modificarHorario">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleHorario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="verificarPerfil" type="s1:verificarPerfil"/>
<xs:complexType name="verificarPerfil">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarCaberaAsistenciaResponse" type="s1:consultarCaberaAsistenciaResponse"/>
<xs:complexType name="consultarCaberaAsistenciaResponse">
  <xs:sequence>
    <xs:element name="idCabeceraAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarDetalleJustificacionResponse" type="s1:consultarDetalleJustificacionResponse"/>
<xs:complexType name="consultarDetalleJustificacionResponse">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDetalleJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarDistribucionResponse" type="s1:consultarDistribucionResponse"/>
<xs:complexType name="consultarDistribucionResponse">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarDocenteResponse" type="s1:consultarDocenteResponse"/>
<xs:complexType name="consultarDocenteResponse">
  <xs:sequence>
    <xs:element name="idDocente" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarEstudianteResponse" type="s1:consultarEstudianteResponse"/>
<xs:complexType name="consultarEstudianteResponse">
  <xs:sequence>
    <xs:element name="apellido" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="nombre" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="consultarTipoAsistenciaResponse" type="s1:consultarTipoAsistenciaResponse"/>
<xs:complexType name="consultarTipoAsistenciaResponse">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarTipoJustificacionResponse" type="s1:consultarTipoJustificacionResponse"/>
<xs:complexType name="consultarTipoJustificacionResponse">
  <xs:sequence>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarAsistenciaResponse" type="s1:ingresarAsistenciaResponse"/>
<xs:complexType name="ingresarAsistenciaResponse">
  <xs:sequence>
    <xs:element name="idEstudiante" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarEventoResponse" type="s1:ingresarEventoResponse"/>
<xs:complexType name="ingresarEventoResponse">
  <xs:sequence>
    <xs:element name="backgroundColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="borderColor" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="detalle" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaFin" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="fechaInicio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idEvento" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarHorarioResponse" type="s1:ingresarHorarioResponse"/>
<xs:complexType name="ingresarHorarioResponse">
  <xs:sequence>
    <xs:element name="dia" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaEntrada" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="horaSalida" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarJustificacionResponse" type="s1:ingresarJustificacionResponse"/>
<xs:complexType name="ingresarJustificacionResponse">
  <xs:sequence>
    <xs:element name="idTipoJustificacion" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idUsuario" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="observacion" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="url" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="iniciarSesionResponse" type="s1:iniciarSesionResponse"/>
<xs:complexType name="iniciarSesionResponse">
  <xs:sequence>
    <xs:element name="contraseña" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="usuario" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarAsistenciaResponse" type="s1:modificarAsistenciaResponse"/>
<xs:complexType name="modificarAsistenciaResponse">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</wsdl:types>

```

```

<xs:element name="consultarCoordinadorResponse" type="s1:consultarCoordinadorResponse"/>
<xs:complexType name="consultarCoordinadorResponse">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="anio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="documento" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estado" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCoordinador" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarBitacora" type="s1:ingresarBitacora"/>
<xs:complexType name="ingresarBitacora">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idObservacion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarBitacoraResponse" type="s1:ingresarBitacoraResponse"/>
<xs:complexType name="ingresarBitacoraResponse">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idObservacion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarCoordinador" type="s1:ingresarCoordinador"/>
<xs:complexType name="ingresarCoordinador">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="anio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="documento" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estado" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarCoordinadorResponse" type="s1:ingresarCoordinadorResponse"/>
<xs:complexType name="ingresarCoordinadorResponse">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="anio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="documento" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estado" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ingresarTiempo" type="s1:ingresarTiempo"/>
<xs:complexType name="ingresarTiempo">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="horas" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCarrera" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minutos" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="segundos" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="ingresarTiempoResponse" type="s1:ingresarTiempoResponse"/>
<xs:complexType name="ingresarTiempoResponse">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="horas" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCarrera" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minutos" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="segundos" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarPersonalizacion" type="s1:modificarPersonalizacion"/>
<xs:complexType name="modificarPersonalizacion">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="modificarPersonalizacionResponse" type="s1:modificarPersonalizacionResponse"/>
<xs:complexType name="modificarPersonalizacionResponse">
  <xs:sequence>
    <xs:element name="idAsistencia" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarBitacora" type="s1:consultarBitacora"/>
<xs:complexType name="consultarBitacora">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idBitacora" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idObservacion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarBitacoraResponse" type="s1:consultarBitacoraResponse"/>
<xs:complexType name="consultarBitacoraResponse">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="fecha" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idBitacora" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDistribucion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idObservacion" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipoAsistencia" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:element name="consultarTiempo" type="s1:consultarTiempo"/>
<xs:complexType name="consultarTiempo">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="horas" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCarrera" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTiempo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minutos" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="segundos" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarTiempoResponse" type="s1:consultarTiempoResponse"/>
<xs:complexType name="consultarTiempoResponse">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="horas" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCarrera" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idPeriodo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTiempo" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="minutos" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="segundos" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="consultarCoordinador" type="s1:consultarCoordinador"/>
<xs:complexType name="consultarCoordinador">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:element name="anio" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="documento" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="estado" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idCoordinador" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idDocente" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idTipo" type="xs:int" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

Figura 145. Código XML para los tipos de datos de los servicios.

```

<wsdl:message name="ingresarHorarioSoapIn">
  <wsdl:part name="parameters" type="ingresarHorario"/>
</wsdl:message>
<wsdl:message name="ingresarHorarioSoapOut">
  <wsdl:part name="parameters" type="ingresarHorarioResponse"/>
</wsdl:message>
<wsdl:message name="ingresarEventoSoapOut">
  <wsdl:part name="parameters" type="tns:ingresarEventoResponse"/>
</wsdl:message>
<wsdl:message name="consultarTipoAsistenciaSoapOut">
  <wsdl:part name="parameters" type="tns:consultarTipoAsistenciaResponse"/>
</wsdl:message>
<wsdl:message name="consultarEventosSoapIn">
  <wsdl:part name="parameters" type="tns:consultarEventos"/>
</wsdl:message>
<wsdl:message name="consultarEventosSoapOut">
  <wsdl:part name="parameters" type="tns:consultarEventosResponse"/>
</wsdl:message>
<wsdl:message name="modificarEventoSoapIn">
  <wsdl:part name="parameters" type="tns:modificarEvento"/>
</wsdl:message>

```



```

<wsdl:message name="verificarPerfilSoapIn">
  <wsdl:part name="parameters" type="tns:verificarPerfil"/>
</wsdl:message>
<wsdl:message name="verificarPerfilSoapOut">
  <wsdl:part name="parameters" type="verificarPerfilResponse"/>
</wsdl:message>
<wsdl:message name="consultarEstudianteSoapIn">
  <wsdl:part name="parameters" type="consultarEstudiante"/>
</wsdl:message>
<wsdl:message name="consultarEstudianteSoapOut">
  <wsdl:part name="parameters" type="consultarEstudianteResponse"/>
</wsdl:message>
<wsdl:message name="consultarTipoAsistenciaSoapIn">
  <wsdl:part name="parameters" type="consultarTipoAsistencia"/>
</wsdl:message>
<wsdl:message name="consultarAsistenciaSoapOut">
  <wsdl:part name="parameters" type="consultarTipoAsistenciaResponse"/>
</wsdl:message>
<wsdl:message name="modificarHorarioSoapIn">
  <wsdl:documentation></wsdl:documentation>
  <wsdl:part name="parameters" type="tns:modificarHorario"/>
</wsdl:message>
<wsdl:message name="modificarHorarioSoapOut">
  <wsdl:part name="parameters" type="tns:modificarHorarioResponse"/>
</wsdl:message>
<wsdl:message name="ingresarEventoSoapIn">
  <wsdl:part name="parameters" type="tns:ingresarEvento"/>
</wsdl:message>
<wsdl:message name="ingresarJustificacionSoapIn">
  <wsdl:part name="parameters" type="ingresarJustificacion"/>
</wsdl:message>
<wsdl:message name="ingresarJustificacionSoapOut">
  <wsdl:part name="parameters" type="ingresarJustificacionResponse"/>
</wsdl:message>
<wsdl:message name="modificarAsistenciaSoapIn">
  <wsdl:part name="parameters" type="modificarAsistencia"/>
</wsdl:message>
<wsdl:message name="modificarAsistenciaSoapOut">
  <wsdl:part name="parameters" type="modificarAsistenciaResponse"/>
</wsdl:message>
<wsdl:message name="ingresarAsistenciaSoapIn">
  <wsdl:part name="parameters" type="ingresarAsistencia"/>
</wsdl:message>
<wsdl:message name="ingresarAsistenciaSoapOut">
  <wsdl:part name="parameters" type="ingresarAsistenciaResponse"/>
</wsdl:message>
<wsdl:message name="consultarJustificacionSoapIn">
  <wsdl:part name="parameters" type="consultarJustificacion"/>
</wsdl:message>
<wsdl:message name="consultarJustificacionSoapOut">
  <wsdl:part name="parameters" type="consultarJustificacionResponse"/>
</wsdl:message>
<wsdl:message name="consultarDetalleJustificacionSoapIn">
  <wsdl:part name="parameters" type="consultarDetalleJustificacion"/>
</wsdl:message>
<wsdl:message name="consultarDetalleJustificacionSoapOut">
  <wsdl:part name="parameters" type="consultarDetalleJustificacionResponse"/>
</wsdl:message>

```

```

<wsdl:message name="iniciarSesionSoapIn">
  <wsdl:part name="parameters" type="tns:iniciarSesion"/>
</wsdl:message>
<wsdl:message name="iniciarSesionSoapOut">
  <wsdl:part name="parameters" type="tns:iniciarSesionResponse"/>
</wsdl:message>
<wsdl:message name="consultarDocenteSoapIn">
  <wsdl:part name="parameters" type="consultarDocente"/>
</wsdl:message>
<wsdl:message name="consultarDocenteSoapOut">
  <wsdl:part name="parameters" type="consultarDocenteResponse"/>
</wsdl:message>
<wsdl:message name="consultarTipoJustificacionSoapIn">
  <wsdl:part name="parameters" type="consultarTipoJustificacion"/>
</wsdl:message>
<wsdl:message name="consultarTipoJustificacionSoapOut">
  <wsdl:part name="parameters" type="consultarHorarioResponse"/>
</wsdl:message>
<wsdl:message name="consultarDistribucionSoapIn">
  <wsdl:part name="parameters" type="consultarDistribucion"/>
</wsdl:message>
<wsdl:message name="consultarDistribucionSoapOut">
  <wsdl:part name="parameters" type="consultarDistribucionResponse"/>
</wsdl:message>
<wsdl:message name="consultarPeriodoSoapIn">
  <wsdl:part name="parameters" type="consultarPeriodo"/>
</wsdl:message>
<wsdl:message name="consultarPeriodoSoapOut">
  <wsdl:part name="parameters" type="consultarPeriodoResponse"/>
</wsdl:message>
<wsdl:message name="consultarHorarioSoapIn">
  <wsdl:part name="parameters" type="consultarHorario"/>
</wsdl:message>
<wsdl:message name="consultarHorarioSoapOut">
  <wsdl:part name="parameters" type="consultarHorarioResponse"/>
</wsdl:message>
<wsdl:message name="consultarCabeceraAsistenciaSoapIn">
  <wsdl:part name="parameters" type="consultarCaberaAsistencia"/>
</wsdl:message>
<wsdl:message name="consultarCabeceraAsistenciaSoapOut">
  <wsdl:part name="parameters" type="consultarCaberaAsistenciaResponse"/>
</wsdl:message>
<wsdl:message name="consultarJustificacionSoapIn">
  <wsdl:part name="parameters" type="consultarJustificacion"/>
</wsdl:message>
<wsdl:message name="consultarJustificacionSoapOut">
  <wsdl:part name="parameters" type="consultarJustificacionResponse"/>
</wsdl:message>
<wsdl:message name="consultarDetalleJustificacionSoapIn">
  <wsdl:part name="parameters" type="consultarDetalleJustificacion"/>
</wsdl:message>
<wsdl:message name="consultarDetalleJustificacionSoapOut">
  <wsdl:part name="parameters" type="consultarDetalleJustificacionResponse"/>
</wsdl:message>

```

```

<wsdl:message name="modificarEventoSoapOut">
  <wsdl:part name="parameters" type="tns:modificarEventoResponse"/>
</wsdl:message>
<wsdl:message name="consultarTiempoSoapIn">
  <wsdl:part name="parameters" type="tns:consultarTiempo"/>
</wsdl:message>
<wsdl:message name="consultarTiempoSoapOut">
  <wsdl:part name="parameters" type="tns:consultarTiempoResponse"/>
</wsdl:message>
<wsdl:message name="consultarBitacoraSoapIn">
  <wsdl:part name="parameters" type="tns:consultarBitacora"/>
</wsdl:message>
<wsdl:message name="consultarBitacoraSoapOut">
  <wsdl:part name="parameters" type="tns:consultarBitacoraResponse"/>
</wsdl:message>
<wsdl:message name="consultarCoordinadorSoapIn">
  <wsdl:part name="parameters" type="tns:consultarCoordinador"/>
</wsdl:message>
<wsdl:message name="consultarCoordinadorSoapOut">
  <wsdl:part name="parameters" type="tns:consultarCoordinadorResponse"/>
</wsdl:message>
<wsdl:message name="ingresarTiempoSoapIn">
  <wsdl:part name="parameters" type="tns:ingresarTiempo"/>
</wsdl:message>
<wsdl:message name="ingresarTiempoSoapOut">
  <wsdl:part name="parameters" type="tns:ingresarTiempoResponse"/>
</wsdl:message>
<wsdl:message name="ingresarBitacoraSoapIn">
  <wsdl:part name="parameters" type="tns:ingresarBitacora"/>
</wsdl:message>
<wsdl:message name="ingresarBitacoraSoapOut">
  <wsdl:part name="parameters" type="tns:ingresarBitacoraResponse"/>
</wsdl:message>
<wsdl:message name="ingresarCoordinadorSoapIn">
  <wsdl:part name="parameters" type="tns:consultarCoordinador"/>
</wsdl:message>
<wsdl:message name="ingresarCoordinadorSoapOut">
  <wsdl:part name="parameters" type="tns:consultarCoordinadorResponse"/>
</wsdl:message>

```

Figura 146. Código XML para los mensajes de las funciones

```

<wsdl:portType name="cabeceraAsistenciaSOAP">
  <wsdl:operation name="consultarCabeceraAsistencia">
    <wsdl:input name="Request" message="tns:consultarCabeceraAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarCabeceraAsistenciaSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="detalleJustificacionSOAP">
  <wsdl:operation name="consultarDetalleJustificacionSOAP">
    <wsdl:input name="Request" message="tns:consultarDetalleJustificacionSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarDetalleJustificacionSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="distribucionSOAP">
  <wsdl:operation name="consultarDistribucionSOAP">
    <wsdl:input name="Request" message="tns:consultarDistribucionSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarDistribucionSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="docenteSOAP">
  <wsdl:operation name="iniciarSesionSOAP">
    <wsdl:input name="Request" message="tns:iniciarSesionSoapIn"/>
    <wsdl:output name="Response" message="tns:iniciarSesionSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="consultarDocenteSOAP">
    <wsdl:input name="Request" message="tns:consultarDocenteSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarDocenteSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="estudianteSOAP">
  <wsdl:operation name="consultarEstudianteSOAP">
    <wsdl:input name="Request" message="tns:consultarEstudianteSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarEstudianteSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="eventosSOAP">
  <wsdl:operation name="ingresarEventoSOAP">
    <wsdl:input name="Request" message="tns:ingresarEventoSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarEventoSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="modificarEventoSOAP">
    <wsdl:input name="Request" message="tns:modificarEventoSoapIn"/>
    <wsdl:output name="Response" message="tns:modificarEventoSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="consultarEventoSOAP">
    <wsdl:input name="Request" message="tns:consultarEventosSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarEventosSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="horarioSOAP">
  <wsdl:operation name="ingresarHorarioSOAP">
    <wsdl:input name="Request" message="tns:ingresarHorarioSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarHorarioSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="modificarHorarioSOAP">
    <wsdl:input name="Request" message="tns:modificarHorarioSoapIn"/>
    <wsdl:output name="Response" message="tns:modificarHorarioSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="consultarHorarioSOAP">
    <wsdl:input name="Request" message="tns:consultarHorarioSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarHorarioSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="justificacionSOAP">
  <wsdl:operation name="ingresarJustificacionSOAP">
    <wsdl:input name="Request" message="tns:ingresarJustificacionSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarJustificacionSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="consultarJustificacionSOAP">
    <wsdl:input name="Request" message="tns:consultarJustificacionSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarJustificacionSoapOut"/>
  </wsdl:operation>
</wsdl:portType>

```

```
<wsdl:portType name="periodoSOAP">
  <wsdl:operation name="consultarPeriodoSOAP">
    <wsdl:input name="Request" message="tns:consultarPeriodoSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarPeriodoSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="tipoAsistenciaSOAP">
  <wsdl:operation name="consultarTipoAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:consultarTipoAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarTipoAsistenciaSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="tipoJustificacionSOAP">
  <wsdl:operation name="consultarTipoJustificacionSOAP">
    <wsdl:input name="Request" message="tns:consultarTipoJustificacionSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarTipoJustificacionSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="perfilSOAP">
  <wsdl:operation name="verificarPerfilSOAP">
    <wsdl:input name="Request" message="tns:verificarPerfilSoapIn"/>
    <wsdl:output name="Response" message="tns:verificarPerfilSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="asistenciaSOAP">
  <wsdl:operation name="consultarAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:consultarTipoAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarAsistenciaSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ingresarAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:ingresarAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarAsistenciaSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="modificarAsistenciaSOAP">
    <wsdl:input name="Request" message="tns:modificarAsistenciaSoapIn"/>
    <wsdl:output name="Response" message="tns:modificarAsistenciaSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
```

```

<wsdl:portType name="tiempoSOAP">
  <wsdl:operation name="consultarTiempoSOAP">
    <wsdl:input name="Request" message="tns:consultarTiempoSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarTiempoSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ingresarTiempoSOAP">
    <wsdl:input name="Request" message="tns:ingresarTiempoSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarTiempoSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="bitacoraSOAP">
  <wsdl:operation name="consultarBitacoraSOAP">
    <wsdl:input name="Request" message="tns:consultarBitacoraSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarBitacoraSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="ingresarBitacoraSOAP">
    <wsdl:input name="Request" message="tns:ingresarBitacoraSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarBitacoraSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="coordinadorSOAP">
  <wsdl:operation name="ingresarCoordinadorSOAP">
    <wsdl:input name="Request" message="tns:ingresarCoordinadorSoapIn"/>
    <wsdl:output name="Response" message="tns:ingresarCoordinadorSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="consultarCoordinadorSOAP">
    <wsdl:input name="Request" message="tns:consultarCoordinadorSoapIn"/>
    <wsdl:output name="Response" message="tns:consultarCoordinadorSoapOut"/>
  </wsdl:operation>
</wsdl:portType>

```

Figura 126. Código XML para las operaciones permitidas en los servicios.

```

<wsdl:binding name="asistenciaBindingSOAP" type="tns:asistenciaSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="modificarAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

<wsdl:binding name="cabeceraAsistenciaBindingSOAP" type="tns:cabeceraAsistenciaSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarCabeceraAsistencia">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="detalleJustificacionBindingSOAP" type="tns:detalleJustificacionSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarDetalleJustificacionSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="distribucionBindingSOAP" type="tns:distribucionSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarDistribucionSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="horarioBindingSOAP" type="tns:horarioSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarHorarioSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarHorarioSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="modificarHorarioSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

<wsdl:binding name="justificacionBindingSOAP" type="tns:justificacionSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarJustificacionSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarJustificacionSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="perfilBindingSOAP" type="tns:perfilSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="verificarPerfilSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="periodoBindingSOAP" type="tns:periodoSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarPeriodoSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="tipoAsistenciaBindingSOAP" type="tns:tipoAsistenciaSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarTipoAsistenciaSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```



```

<wsdl:binding name="tipoJustificacionBindingSOAP" type="tns:tipoJustificacionSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarTipoJustificacionSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="docenteBindingSOAP" type="tns:docenteSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarDocenteSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="iniciarSesionSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="estudianteBindingSOAP" type="tns:estudianteSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarEstudianteSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

<wsdl:binding name="eventosBindingSOAP" type="tns:eventosSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="modificarEventoSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="consultarEventoSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarEventoSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="bitacoraBindingSOAP" type="tns:bitacoraSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarBitacoraSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarBitacoraSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

<wsdl:binding name="tiempoBindingSOAP" type="tns:tiempoSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarTiempoSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarTiempoSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="coordinadorBindingSOAP" type="tns:coordinadorSOAP">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="consultarCoordinadorSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ingresarCoordinadorSOAP">
    <soap:operation soapAction="http://www.sampleAction.com" style="document"/>
    <wsdl:input name="Request">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="Response">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

Figura 147. Código XML para especificación de protocolos de comunicación

```

<wsdl:service name="asistenciaService">
  <wsdl:port name="asistenciaPortSOAP" binding="tns:asistenciaBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/asistencia.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="cabeceraAsistenciaService">
  <wsdl:port name="cabeceraPortSOAP" binding="tns:cabeceraAsistenciaBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/cabeceraAsistencia.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="docenteService">
  <wsdl:port name="docentePortSOAP" binding="tns:docenteBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/docente.asmx"/>
  </wsdl:port>
</wsdl:service>

```

```

<wsdl:service name="estudianteService">
  <wsdl:port name="estudiantePortSOAP" binding="tns:estudianteBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/estudiante.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="distribucionService">
  <wsdl:port name="distribucionPortSOAP" binding="tns:distribucionBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/distribucion.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="detalleJustificacionService">
  <wsdl:port name="detalleJustificacionPortSOAP" binding="tns:detalleJustificacionBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/detalleJustificacion.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="eventosService">
  <wsdl:port name="eventosPortSOAP" binding="tns:eventosBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/eventos.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="horarioService">
  <wsdl:port name="horarioPortSOAP" binding="tns:horarioBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/horario.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="justificacionService">
  <wsdl:port name="justificacionPortSOAP" binding="tns:justificacionBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/justificacion.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="perfilService">
  <wsdl:port name="perfilPortSOAP" binding="tns:perfilBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/perfil.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="periodoService">
  <wsdl:port name="periodoPortSOAP" binding="tns:periodoBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/periodo.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="tipoAsistenciaService">
  <wsdl:port name="tipoAsistenciaPortSOAP" binding="tns:tipoAsistenciaBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/tipoAsistencia.asmx"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="tipoJustificacionService">
  <wsdl:port name="tipoJustificacionPortSOAP" binding="tns:tipoJustificacionBindingSOAP">
    <soap:address location="http://localhost/CASYS/Services/tipoJustificacion.asmx"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
<wsdl:service name="bitacoraService">
  <wsdl:port name="bitacoraPortSOAP" binding="tns:bitacoraBindingSOAP">
    <soap:address location="www.exampleLocation.com/Port1"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="tiempoService">
  <wsdl:port name="tiempoPortSOAP" binding="tns:tiempoBindingSOAP">
    <soap:address location="www.exampleLocation.com/Port1"/>
  </wsdl:port>
</wsdl:service>
<wsdl:service name="coordinadorService">
  <wsdl:port name="coordinadorPort" binding="tns:coordinadorBindingSOAP">
    <soap:address location="www.exampleLocation.com/Port1"/>
  </wsdl:port>
</wsdl:service>

```

Figura 148. Código XML correspondiente a los servicios Web

ANEXO 4

MANUAL DE USUARIO DEL SISTEMA.

Manual de usuario CASYS

(Sistema de control de asistencia)

Introducción

CASYS es el sistema para llevar el control de registro de las asistencias de los estudiantes que asisten en los cursos regulares de las distintas carreras de la ESPAM MFL. El propósito de este manual es facilitar al usuario el entendimiento del manejo correcto del sistema en los diferentes módulos a los cuales tengan acceso a través de capturas de pantallas y pasos a seguir.



Inicio de sesión

CASYS está integrado con la base de datos del sistema integral de la ESPAM MFL por lo que para ingresar al sistema lo podrán hacer con su usuario y contraseña regular de la institución.

El sistema está dividido en 4 partes, correspondientes a cada tipo de usuario (docente, coordinador, administrador y directores de carreras), cada uno de estos usuarios tendrá el acceso a ciertas partes específicas del sistema y permisos para poder o no manipular la información de los mismos.

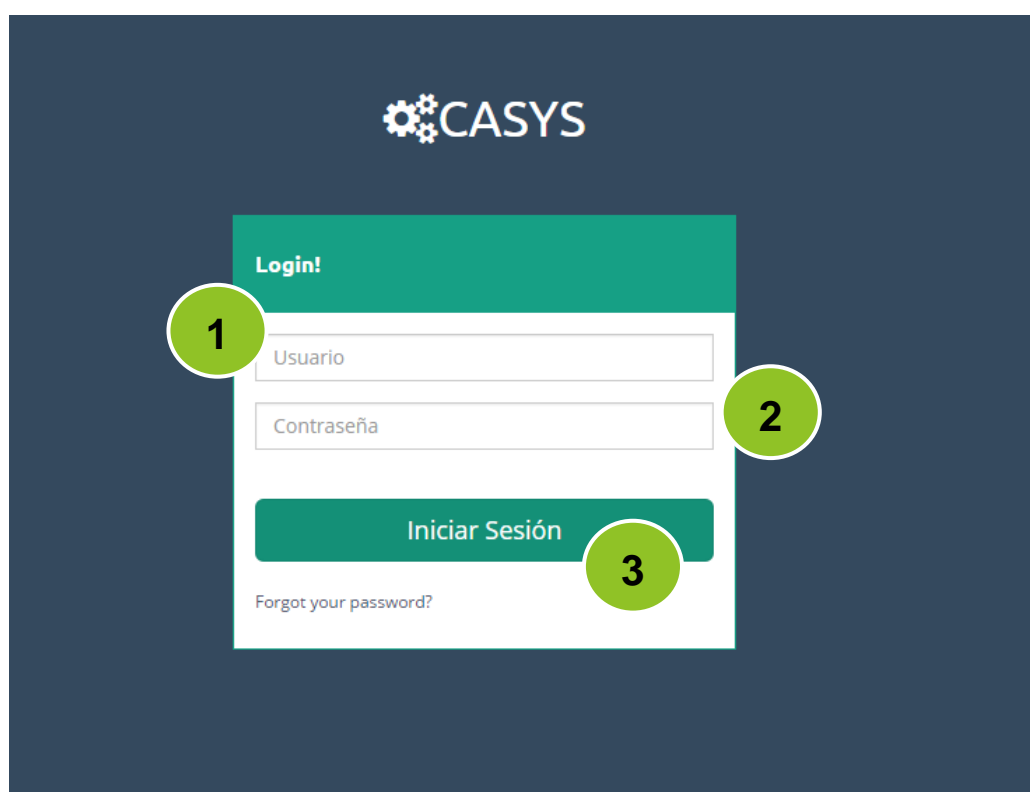


Figura 1. Página de ingreso de CASYS.

- 1.- Ingresar el usuario (el mismo usuario regular de la institución).
- 2.- Ingresar la contraseña (la misma contraseña regular de la institución).
- 3.- Dar click en el botón “Iniciar Sesión”.

Una vez usted haya hecho todos los pasos anteriores podrá visualizar la pantalla de inicio de CASYS.



Figura 2. Página de inicio de CASYS (Modulo Administrador).

- 1.- Menú de los módulos a los cuales tendrá acceso.
- 2.- El cronometro que muestra el tiempo de navegación que usted lleva dentro del sistema. Este tiempo será almacenado para propósitos estadísticos en el momento de cerrar la sesión.
- 3.- Menú de cerrar sesión.

Docentes

Dentro de CASYS los docentes como usuarios tendrán acceso a varios módulos los cuales son los siguientes:

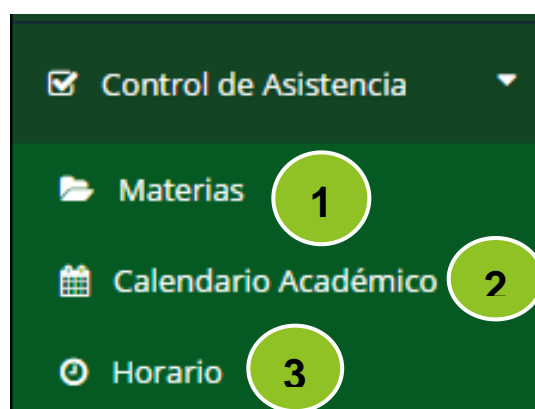


Figura 3. Menú (Modulo Docente).

Módulo de Materias:

Dentro de este módulo usted va a encontrar la lista de las materias que tiene disponible a su cargo (las que se encuentran entre llaves figura 4), cada una de estas posee en su estructura interna 3 categorías más las cuales se detallan a continuación.

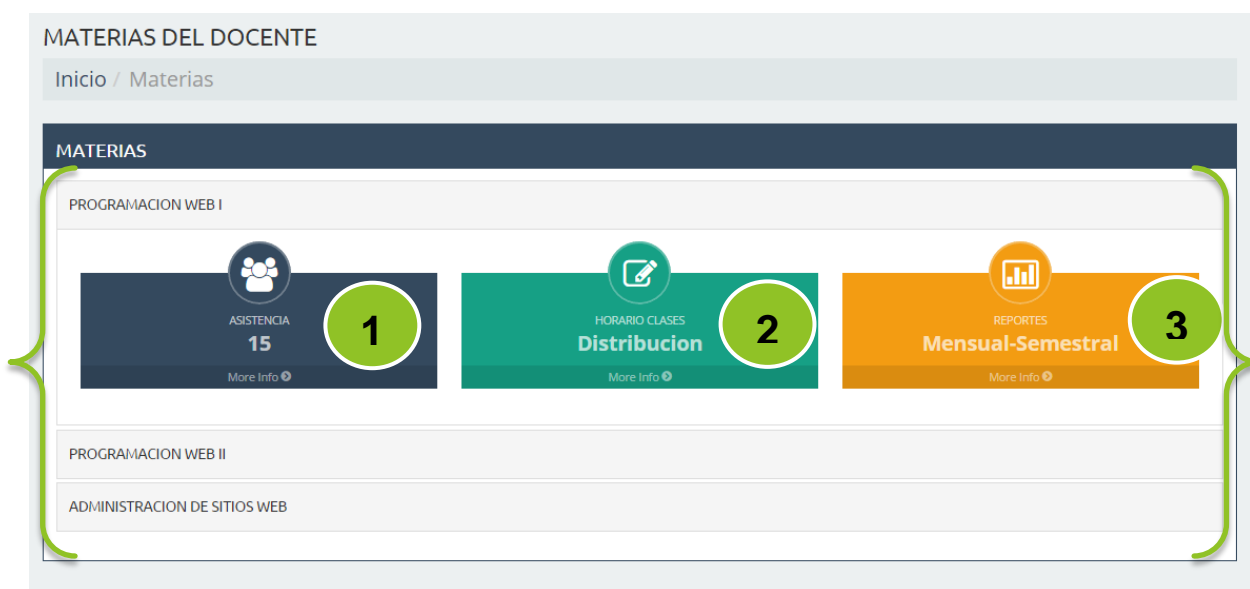


Figura 4. Módulo de materias.

- 1.1. **Módulo de asistencias:** Sirve para tomar las asistencias de los estudiantes, para poder acceder a esta página deberá de tener ingresado los horarios de clases tal como se indica en la sección de horario (ir a horario). En el caso de que ya tenga ingresado sus horarios de clases le aparecerá la siguiente página en la que le aparecerán los detalles generales del curso (figura 5) como son el semestre, paralelo, nombre de la materia y el nombre del docente, el horario de clases de esa materia y otros datos estadísticos, además en la parte inferior del detalle aparecerá un botón para habilitar la toma de las asistencias del día y la lista de estudiantes matriculados en esa materia.

Asistencia CARRERA DE INFORMÁTICA

Inicio / Distribuciones / Asistencia

INFORMACIÓN GENERAL

DATOS DEL CURSO

Docente:	PINARGOTE BRAVO VICTOR JOEL
Curso:	PROGRAMACION WEB I
Semestre:	6
Paralelo:	A

HORARIO DE CLASES

Dia	Entrada	Salida	Horas
LU	17:00	19:00	2
MA	20:00	22:00	2
MI			0
JU			0
VI			0

RESUMEN SEMESTRAL:

Total horas del semestre:	64
Total de mes del mes anterior:	0
Total de asistencia de este mes:	0
Total de horas por semanas:	4

Tomar Asistencia

1

Figura 5. Detalles (Módulo de asistencias).

1.1.1. Tomar asistencia: El botón de tomar asistencias únicamente va a habilitar la toma de la asistencia si en el horario coincide con el día en el que está tomando asistencia, si ese es el caso, la lista de estudiantes se habilitara una casilla para poder empezar a tomar las asistencias la misma que estará sombreada de un color más oscuro para poderla diferenciar (Figura 6, número 1), además en la parte final de la lista se habilitara un botón el cual será el que permitirá confirmar y guardar las asistencias de los estudiantes (figura 6, número 3). También en la parte inferior aparecerá una casilla de verificación.

Lista de Estudiantes		TOTAL ASISTENCIA												PORCENTAJE															
N	Nomina Estudiantes	Día						Julio						ANTERIOR			ACUMULADA			Julio			ACUMULADA						
		2	3	9	1	7	23	7	7	7	7	7	7	Julio	FI	FI	A	FJ	FI	A	FJ	FI	%A	%FJ	%FI	%A	%F	%TOTAL	
		Horas	0	0	0	0	0	0	0	0	0	0	0	A	FJ	FI	A	FJ	FI	A	FJ	FI							
1	BALLEN FAUBLA CHRISTIAN ALEXIS	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
2	BUSTE ZAMBRANO JOSE ORLANDO	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
3	CEDEÑO MENDOZA TITO ARGENIS	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
4	DELGADO ZAMBRANO PABLO RICARDO	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
5	GILCES VERA ELIANA LISBETH	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
6	LOOR LOOR LUIS ANTONIO	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
7	MARCILLO CEDEÑO ADOLFO ANTONIO	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
8	MENDOZA LOOR EDDY GREGORIO	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
9	MENENDEZ CHAVEZ HECTOR MIGUEL	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
10	MORA SALTOS MARIANA LICETH	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
11	MUÑOZ LOOR CAROLINA LISBETH	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
12	NAVARRRETE RODRIGUEZ EDWIN GERMAN	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
13	PAZMIÑO PALMA MARIA GABRIELA	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
14	PINARGOTE CARVAJAL RAQUEL KAROLINA	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
15	PINARGOTE SANTANA JOSE LUIS	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
16	RIVERA VERA JONIV JAVIER	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
17	SANCHEZ MACIAS CINTHIA MABEL	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
18	SANTANA MONTEDEOCA MICHAEL JHONNY	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%
19	SECARA MORA JOSE ISAAC	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	2	0%	0%	0%	4%	0%	4%

Acepto que la información fue verificada y está apegada a la verdad, confirmo que puede utilizada uso conveniente de la coordinación académica.

2

3

Confirmar Asistencia

Figura 6. Lista de estudiantes (Módulo de asistencias).

1.1.2. Aceptar términos (casilla de verificación): Esta casilla de verificación se utiliza para habilitar el permiso de impresión de reportes de las distintas materias a los coordinadores de año, esto se deberá de activar a inicio de cada mes luego de que se haya terminado el mes anterior de asistencias. Es decir, supongamos que se ya se han colocado las asistencias del mes de abril, para poder dar permiso a un coordinador de imprimir los reportes de ese mes de abril se debería de aceptar la casilla de verificación en el mes siguiente, es decir, en mayo (figura 6, número 2).

CARRERA DE INFORMÁTICA PROGRAMACION WEB I

Inicio / Materias / Horario Individual

HORARIO DE CLASES 3

Semana Hoy

	Lun	Mar	Mi	Ju	Vi
17:00	17:00 PROGRAMACION WEB I				
18:00					
19:00					
20:00		20:00 PROGRAMACION WEB I			
21:00					
22:00					

Figura 7. Horario individual (Módulo de asistencias).

1.1.3. Horario individual: Se mostrara el horario individual de cada una de las materias con sus respectivas distribuciones de horas.

Reportes PROGRAMACION WEB I 4

Inicio / Materias / Reportes

Reporte COMBO COMBO

30 DE MARZO - 27 DE AGOSTO 2015 (IN) Seleccione el mes Reporte Mensual Reporte General

Seleccione el mes

- Marzo - 2015
- Abril - 2015
- Mayo - 2015
- Junio - 2015
- Julio - 2015
- Agosto - 2015


DATOS DE CURSO

 DOCENTE: VICTOR JOEL PINARGOTE BRAVO
 CURSO: PROGRAMACION WEB II
 SEMESTRE: 7 PARALELO: A
 PERIODO: 30 DE MARZO - 27 DE AGOSTO 2015
 (INICIO DE CLASES LUNES 13 DE ABRIL)

HORARIO DE CLASES

DÍA	ENTRADA	SALIDA	HORAS
LU			0
MA	19:00	21:00	2
MI	17:00	19:00	2
JU			0
VI			0

RESUMEN SEMESTRAL

Total acumulado del semestre:	4
Total de mes del mes anterior:	0
Total de asistencia de este mes:	4
Total de hora en 16 semanas:	4
Total de horas no trabajadas:	0
Total de horas por semanas:	4

N°	Nómina de Estudiantes	Día		TOTAL ASISTENCIA											PORCENTAJE							
		Meses	Horas	ANTERIOR											L	Julio			ACUMULADA/TOTAL			
				7	7	Julio	ANTERIOR			ACUMULADA			% ASIS	% FJ		% FI	% ASIS	% FALTAS	% TOTAL			
1	ACOSTA ZAMBRANO NIXON EDUARDO	2	2	0	2	0	2	0	0	0	0	2	0	0	2	4	0	0	0	0	0	100
2	ALAVA CAGUA NATHALY VALERIA	2	0	2	0	2	0	0	0	0	2	0	0	2	4	0	0	0	0	0	0	100
3	BAILON DELGADO DAYANA HELEN	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
4	CEDEÑO REYES MARIA BELEN	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
5	CEVALLOS MOLINA KARLA ESTEFANIA	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
6	COBEÑA CEDEÑO SINDY MONSERRAT	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
7	CONFORME ZAMBRANO JOHANNA STEPHANIE	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
8	DIAZ ENCARNACION MONICA LISSETH	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
9	FARIAS CHICA LUISA KATERINE	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
10	GARCIA GUERRERO BRYAN FERNANDO	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
11	MENDOZA MENDIETA MARIA PAOLA	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
12	MERA CANTOS JOSE SIMON	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
13	MORRILLO BRAVO ANDERSON GUSTAVO	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
14	MUÑOZ NAVARRETE GEMA JAHAIRA	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100
15	MURILLO MONTESDEOCA JENIFFER RAQUEL	2	2	4	0	0	0	0	0	0	4	0	0	4	4	100	0	0	0	100	0	100

Figura 8. Reportes (Módulo de asistencias).

- 1.1.1. Reportes:** Podrá obtener los reportes de asistencias de cada una de las materias y de cada uno de los los periodos respectivos, en primer lugar deberá de escoger el periodo (combo numero 1) y posteriormente podrá escoger de entre los meses (combo numero 2) por los cuales se encuentra compuesto ese periodo de clases.

Módulo Calendario

Dentro del módulo calendario el docente será capaz de observar cuales son los eventos principales que se tienen planificados a lo largo del periodo de clases (figura 9.2), no se le tendrá permitido modificar ningún evento de los que se encontrarán en el calendario. Existirán colores reservados para especificar eventos de inicio y fin de clases, así como también para determinar el inicio de matrículas normal y extraordinario (figura 9.1).

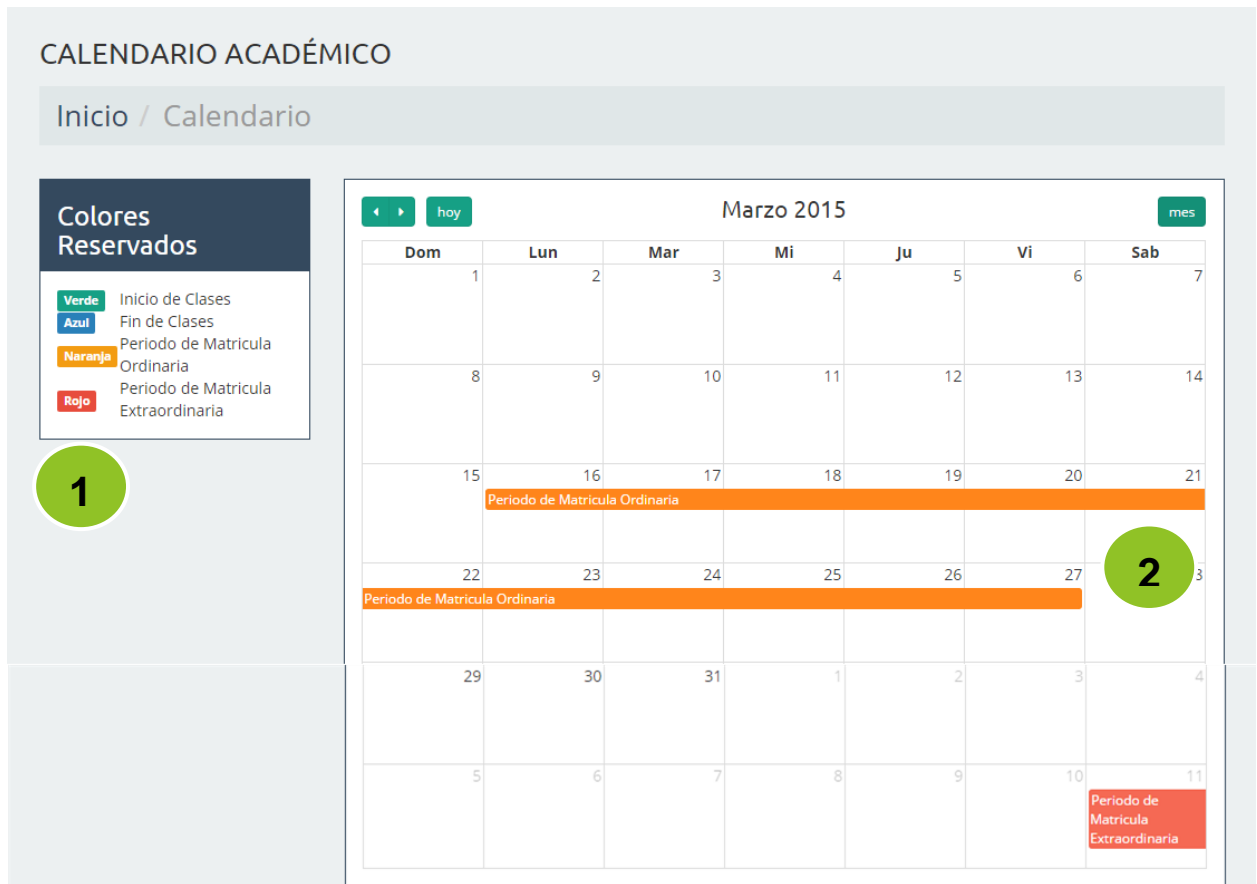


Figura 9. Calendario (Módulo de calendario).

Módulo Horario



Figura 10. Horario general de clases (Módulo de horario).

Este es un modulo general para todas las materias, aquí usted debera armar su horario de acuerdo a la distribucion asignada en sus horas de trabajo. en la parte derecha del módulo le aparecera un recuadro con todas las materias a las cuales el docente imparte clases (figura 10.1), cada una de estas materias usted las podra arrastrar hacia el horario como tal y ubicarlo en el dia y horas que le corresponde (figura 10.2). Además usted podra expandir a la materia según el numero de horas que tenga cada dia con esa materia (figura 10.3) y modificar el dia arrastrando la materia dentro del lugar deseado en el horario.

Directores de carreras

Dentro del módulo de directores de carreras además de los accesos a los cuales tendrán acceso por ser docente, se le agregara adicional una opción que será la de justificación. Dicho módulo de rústicación servirá para justificar las inasistencias de los estudiantes. (Figura 11)

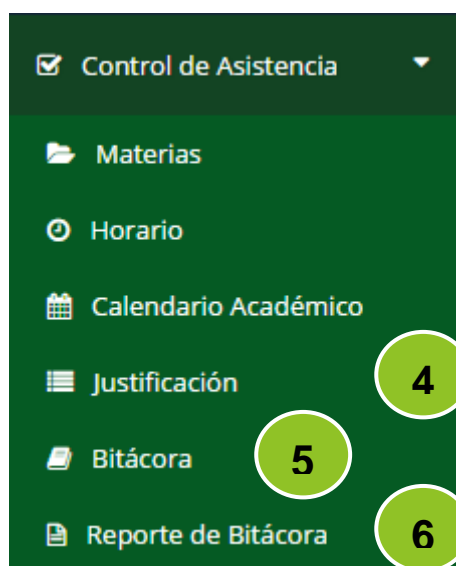


Figura 11. Menú Director de Carrera.

Justificación

El módulo de justificación estará conformado por 2 partes:

JUSTIFICACIÓN DIRECCIÓN

Inicio / Justificación

1

JUSTIFICACIÓN

DESDE: dd/mm/aaaa HASTA: dd/mm/aaaa TIPO: Normal CÉDULA:

DATOS DEL ALUMNO IN-ASISTENCIAS A CLASES

2

ASISTENCIAS EXTRAORDINARIAS

DESDE INICIO DE CLASES: 1/1/1 HASTA: dd/mm/aaaa TIPO: Especial CÉDULA:

DATOS DEL ALUMNO MATERIAS

Figura 13. Paneles del módulo de justificación.

2.1. **Justificación:** El panel de justificación el director de carrera deberá de ingresar el rango de fechas las cuales va a justificar, para ello, el rango de fechas que se pueda justificar dependerá únicamente del tipo de justificación que realizara, por ejemplo, si desea ejecutar una justificación de tipo NORMAL el rango de fechas actual no podrá ser mayor a 2 días (48 horas) después del día de la falta, por lo contrario si el tipo de justificación es de tipo ESPECIAL esa limitación de 48 horas no se aplicara.

JUSTIFICACIÓN

DESDE: dd/mm/aaaa HASTA: dd/mm/aaaa TIPO: Normal CÉDULA:

DATOS DEL ALUMNO IN-ASISTENCIAS A CLASES

Calendario: julio de 2015

lu.	ma.	mi.	ju.	vi.	sá.	do.
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Figura 14. Panel de justificación.

Luego de escoger las fechas correctamente y el tipo de justificación que desea realizar, deberá de ingresar las cedula del estudiante a quien le va a justificar las inasistencias a clases y dar click en el botón de buscar para observar las inasistencias de ese estudiante. En el caso de que el estudiante no presente ninguna inasistencia se le presentara una tabla con los datos del estudiante y junto consigo los datos de la tabla detallando que no se posee ninguna asistencia (figura15).

JUSTIFICACIÓN

DESDE: 15/07/2015 HASTA: 15/07/2015 TIPO: Normal CÉDULA: 1312661778 **Buscar!**

DATOS DEL ALUMNO

NOMBRES Y PELLIDOS:
ANZULES REYNA JONATHAN ALEXANDER

SEMESTRE:
4

CARRERA:
ADMINISTRACIÓN DE EMPRESAS

IN-ASISTENCIAS A CLASES

No posee inasistencias

Figura 15. Panel de justificación (datos de alumnos e inasistencias)

Por el caso contrario, si el estudiante si posee inasistencias se le presentara una tabla en donde se muestra la materia a la que ha faltado a clases, la fecha y las horas que ha faltado y un botón para justificar las asistencias (figura 16).

IN-ASISTENCIAS A CLASES

Materia	Fecha/dias faltados	Horas
PROGRAMACION WEB I	08/06/2015	2
PROGRAMACION WEB I	09/06/2015	2

Justificar asistencias!

Figura 16. Tabla con las inasistencias de un alumno.

Luego de presionar sobre el botón de justificar asistencias aparecerá una pequeña pantalla en la que se deberá subir un archivo justificante en donde se detalla la razón de la justificación en conjunto con una observación detallando el porqué de la misma.



Panel de Justificación

Subir Archivo

Justificación.pdf

Seleccionar archivo No se eligió archivo

Subir archivo justificante

Observaciones:

Limit 250 characters.

CANCELAR CONFIRMAR!

Figura 17. Panel de Justificación (subir archivos).

2.2. Asistencias extraordinarias: Las asistencias extraordinarias son de uso para un caso especial, este panel se lo utiliza para colocarle las asistencias a los estudiantes que se han matriculado fuera de los periodos de matriculación, por ejemplo si un estudiante se matriculo una semana después del último día de los periodos normales de matriculación, se le podrá ingresar asistencias pero únicamente desde el inicio de clases hasta el último día del periodo de matrícula extraordinaria, por lo que tendrá inasistencias por el resto de días que estén fuera de ese rango de fechas.

Bitácora del docente

La bitácora del docente sirve para colocar las inasistencias de un docente con una materia, el director deberá de escoger el docente, la materia a la cual desea colocar la falta, la fecha, el tipo de asistencia y una observación en la que se detalle quien genero esa asistencia o cualquier otro detalle personalizado figura (18).

Registro de Actividad INFORMÁTICA

Inicio / Bitácora del Docente

Registro de Actividades

DOCENTES:

MATERIAS:

FECHA:

TIPO DE ASISTENCIA:

OBSERVACIÓN:

Figura 18. Bitácora del Docente.

Toda actividad agrega se lo mostrara en una tabla en la parte inferior en la cual se detalla los datos de esa actividad figura (19).

Registro de Actividades					
Nro	Materia	Periodo	Tipo	Fecha	Observaciones
No tiene registro de actividad					

Figura 19. Bitácora del Docente (tabla de registro de actividades).

Reporte de bitácora

Dentro de este reporte se encarga de visualizar las horas, trabajas, justificadas e injustificadas que un docente ha hecho a lo largo de un semestre, para poder generar un reporte el director deberá de escoger el periodo en primera instancia, posteriormente el semestre y por último el paralelo, una vez hecho todo ese proceso podrá a mandar a generar un reporte el cual se vera de la siguiente manera (figura 20).

Reportes de horas INFORMÁTICA


Inicio / Reportes de Horas

Reporte

SELECCIONE EL PERIODO DESE. SELECCIONE UN SEMESTRE SELECCIONE UN

Reporte Semestral Reporte General

1 de 1 Buscar | Siguiente


 ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ
 CARRERA DE INFORMÁTICA
 SEMESTRE 1 A
 PERIODO 30 DE MARZO - 27 DE AGOSTO 2015 (INICIO DE CLASES LUNES 13 DE ABRIL)
 REGISTRO DE HORAS

Docente	Materia	Semestre	Horas		
			Trabajadas	Justificadas	Injustificadas
AURA DOLORES ZAMBRANO RENDON	CALCULO I	1 A	0	0	0
ENRRY JOSE COX FIGUEROA	FISICA I	1 A	0	0	0
	ALGEBRA LINEAL	1 A	0	0	0
GUILLEMO INTRIAGO CEDEÑO	COMUNICACION TECNICA	1 A	0	0	0
JOSE PATRICIO MUÑOZ MURILLO	QUIMICA Y ESTRUCTURA DE MATERIALES	1 A	0	0	0
	PRACTICAS DE SISTEMAS INFORMATICOS I	1 A	0	0	0
LUIS CRISTOBAL CEDEÑO VALAREZO	LOGICA MATEMATICA	1 A	0	0	0
ORLANDO AYALA PULLAS	GEOMETRIA ANALITICA	1 A	0	0	0

22/07/2015 9:58:37 1

Figura 20.Bitacora del docente (Reporte).

Coordinadores

El coordinador de año tendrá un módulo especial para el también, dentro del mismo el coordinador podrá generar los reportes de asistencias de los semestres a su cargo y de sus materias respectivas.

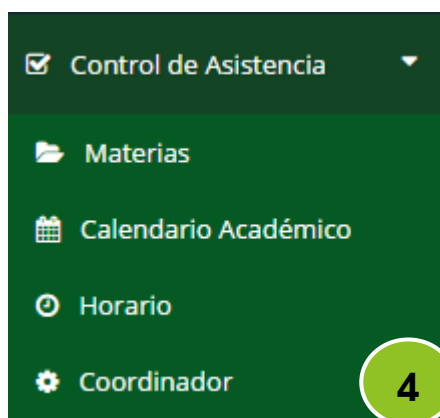


Figura 21. Menú Coordinador.

Coordinador

Dentro de este módulo deberá de escoger el semestre y posteriormente la materia y mandar a generar el reporte (figura 22).

COORDINADOR REPORTES

Inicio / Reportes del coordinador

DATOS

SEMESTRES A CARGO

MATERIAS

Generar Reporte

Figura 22. Módulo coordinador (reportes).

Administradores

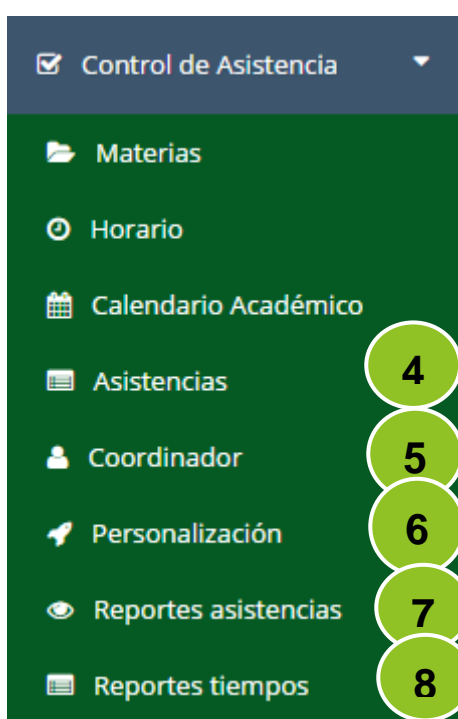


Figura 23. Menú Administrador.

Asistencias

Dentro de asistencias el administrador podrá ingresar las asistencias a los estudiantes de toda la universidad para aquellos docentes que no han ingresado asistencias el día anterior (figura 24).

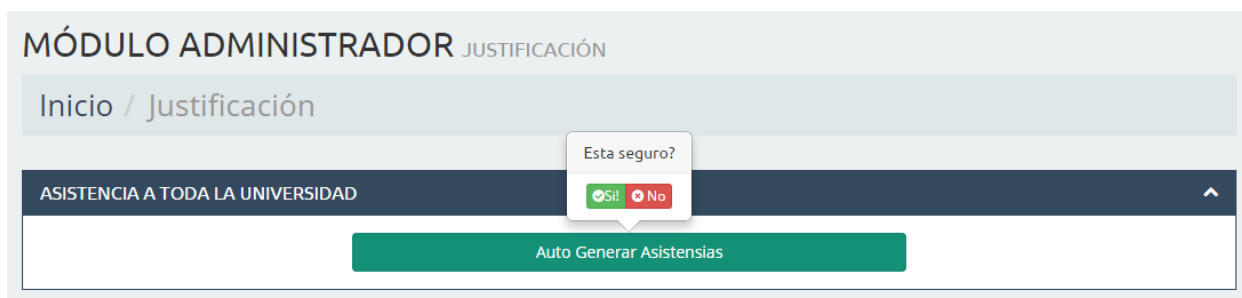


Figura 24. Módulo administrador (Asistencias)

Coordinador

El administrador podrá ingresar a los docentes que serán coordinadores de años respectivos con cada uno de sus carreras y al año al cual será asignado su coordinación (figura 25). Para agregar un docente como coordinador de año deberá de colocar la cedula o pasaporte en el campo buscar y posteriormente se habilitara los campos para poder ingresar a ese docente como coordinador caso contrario aparecerá un mensaje en el que indica que el docente ya se encuentra asignado dentro de los coordinadores.



Figura 25. Módulo administrador (coordinadores)

En la parte inferior se podrá visualizar cuales son los docentes que se encuentran activos y cuales se encuentran desactivados para sus labores como coordinador de año (figura 26).

Coordinadores de año					
Cédula	Nombres	Carrera	Año	Controles	
1302551583	ANGELA CRUZ PINARGOTE COLAMARCO	INFORMÁTICA	QUINTO	Desactivar	Modificar
1301932156	DANIEL MERA MARTINEZ	INFORMÁTICA	PRIMER	Desactivar	Modificar
1306419795	ANGELA LORENA CARREÑO MENDOZA	INFORMÁTICA	SEGUNDO	Desactivar	Modificar
1308648565	ANGEL ALBERTO VELEZ MERO	INFORMÁTICA	PRIMER	Desactivar	Modificar

Coordinadores de año Desactivados					
Cédula	Nombres	Carrera	Año	Controles	
1309563607	JOSE PATRICIO MUÑOZ MURILLO	INFORMÁTICA	PRIMER	Activar	Modificar
1707375489	ORLANDO AYALA PULLAS	INFORMÁTICA	TERCER	Activar	Modificar
1309320685	MARLON RENE NAVIA MENDOZA	INFORMÁTICA	SEGUNDO	Activar	Modificar

Figura 26. Módulo administrador (coordinadores activos y desactivados)

Personalización

Este módulo sirve para colocar el número de semanas por las cuales se encontrara compuesta el periodo normal de clase (figura 27).

Personalizacion del Sistema

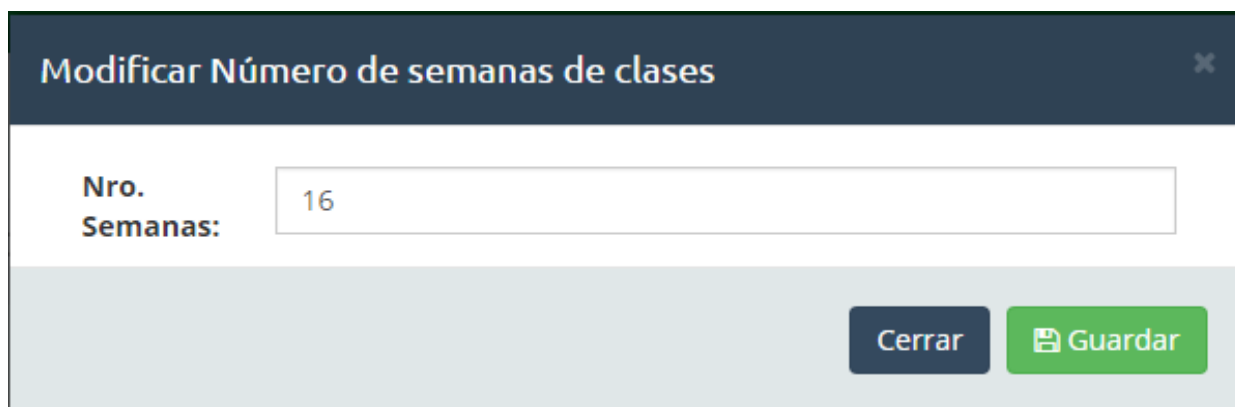
Inicio / Personalización

Número de semanas ▼

[✎ Editar](#)

Figura 27. Módulo administrador (personalización)

Para poder editar el número de semanas simplemente basta con dar click sobre el botón “Editar” y aparece una nueva ventana en la que podrá modificar el número de semanas (figura 28).



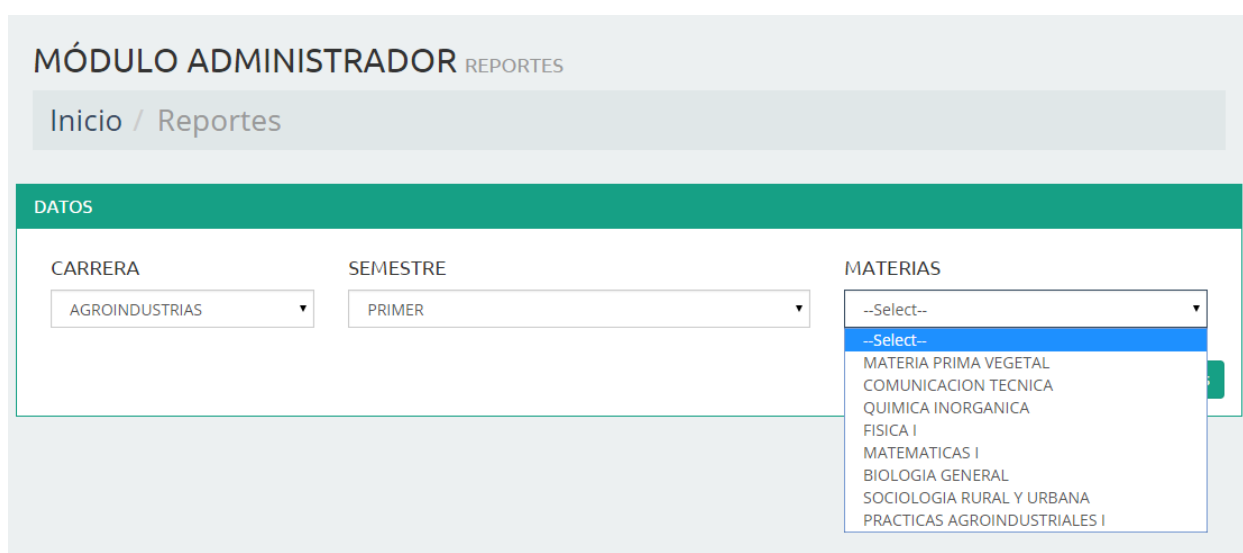
Modificar Número de semanas de clases ✕

Nro. Semanas:

Cerrar **Guardar**

Figura 28. Módulo administrador (Editar)

Reportes asistencias



MÓDULO ADMINISTRADOR REPORTES

Inicio / Reportes

DATOS

CARRERA

SEMESTRE

MATERIAS

 --Select--
 MATERIA PRIMA VEGETAL
 COMUNICACION TECNICA
 QUIMICA INORGANICA
 FISICA I
 MATEMATICAS I
 BIOLOGIA GENERAL
 SOCIOLOGIA RURAL Y URBANA
 PRACTICAS AGROINDUSTRIALES I

Figura 28. Módulo administrador (Editar)

Reportes tiempos

El reporte de tiempo sirve para determinar el tiempo de uso que los docentes le dan a CASYS, para poder generar este reporte deberá de escoger de entre los periodos y posteriormente la carrera (figura 30) y aparecerá una lista con todos los docentes de la carrera y el tiempo de uso que le dan CASYS.

Reportes de tiempo

Inicio / Reportes de tiempo de uso del sistema

Reporte

30 DE MARZO - 27 DE AGOSTO 2015 (INICIO DE CLASES LUNES 13 DE ABRIL)

INFORMÁTICA

Generar Reporte

1 de 1 Buscar | Siguiente



ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ
CARRERA DE INFORMÁTICA
PERIODO 30 DE MARZO - 27 DE AGOSTO 2015 (INICIO DE CLASES LUNES 13 DE ABRIL)
REGISTRO DE TIEMPO DE USO DE CASYS

DOCENTES	HORAS	MINUTOS	SEGUNDOS
VICTOR JOEL PINARGOTE BRAVO	0	92	29
JESSICA JOHANNA MORALES CARRILLO	2	35	60
ANGEL ALBERTO VELEZ MERO	0	24	32

22/07/2015
9:29:07

1

Figura 30. Módulo Administrador (Reportes de tiempo)