



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

CARRERA INFORMÁTICA

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

TEMA:

**SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE
PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL**

AUTORAS:

**GEMA TERESA MENDOZA VERA
JENIFFER MERCEDES PÁRRAGA VERA**

TUTOR

ING. ÁNGEL ALBERTO VÉLEZ MERO, MGS

CALCETA, AGOSTO 2015

DERECHOS DE AUTORÍA

Gema Teresa Mendoza Vera y Jeniffer Mercedes Párraga Vera, 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.

.....
GEMA T. MENDOZA VERA

.....
JENIFFER M. PÁRRAGA VERA

CERTIFICACIÓN DE TUTOR

Ángel Alberto Vélez Mero, certifica haber tutelado la tesis **SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL**, que ha sido desarrollado por Gema Teresa Mendoza Vera y Jeniffer Mercedes Párraga Vera, 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, MGS.

APROBACIÓN DEL TRIBUNAL

Los suscritos, integrantes del tribunal correspondiente, declaran que han **APROBADO** la tesis **SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL**, que ha sido propuesta, desarrollada y sustentada por Gema Teresa Mendoza Vera y Jeniffer Mercedes Párraga Vera, 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. HIRAI DA M. SANTANA CEDEÑO, MGS.
MIEMBRO

.....
ING. AURA D. ZAMBRANO RENDÓN, MGS.
MIEMBRO

.....
LIC. JOSÉ G. INTRIAGO CEDEÑO, MGS.
PRESIDENTE

AGRADECIMIENTO

La vida es el don maravilloso que Dios regala a cada uno de nosotros, más aún, cuando se está rodeada de seres que te motivan a seguir adelante por el camino de la felicidad y el éxito.

Conocedoras que el agradecimiento es uno de los valores que enaltecen a las personas, a través del trabajo de titulación agradecemos a Dios por darnos sabiduría, iluminar nuestros pensamientos para culminar con éxito este valioso trabajo.

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

Al tutor de tesis Ing. Ángel Alberto Vélez Mero, sin su ayuda, orientación, dedicación y motivación no hubiera sido posible la elaboración de este trabajo.

De manera especial al Ing. Lenin Vera Montenegro e Ing. Leonardo Vera Macías y a todo el equipo de profesionales dotados de los más grandes valores y virtudes de la Carrera de Agrícola de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, a todos ellos nuestro leal agradecimiento.

LAS AUTORAS

DEDICATORIA

A Dios quién supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento.

Con todo mi cariño y amor para ti mamá que hiciste todo para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba. A ti papá que desde el cielo me das fuerzas para no dejarme vencer, y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí.

A todas las personas que de una u otra manera estuvieron apoyándome en cada momento de mi vida.

.....
GEMA T. MENDOZA VERA

DEDICATORIA

A Dios por regalarme el don de vida y el don del conocimiento; iluminando cada paso de mi vida, permitiéndome llegar hasta este momento tan importante de mi formación profesional.

A mis padres, por brindarme ejemplos dignos de superación y entrega, gracias a ustedes, hoy puedo alcanzar mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, a ustedes mi eterno agradecimiento, es un privilegio ser su hija.

A mi hermano Carlos Luis, por su apoyo y comprensión en mis esfuerzos de superación profesional. A ti Matteo José, mi angelito, me hubiese encantado compartir este logro junto a ti.

A mis amigos y profesores, por compartir gratos momentos junto a mí, a ellos mis más sinceros agradecimientos.

.....
JENIFFER M. PÁRRAGA VERA

CONTENIDO GENERAL

DERECHOS DE AUTORÍA	ii
CERTIFICACIÓN DE TUTOR	iii
APROBACIÓN DEL TRIBUNAL	iv
AGRADECIMIENTO	v
DEDICATORIA	vi
DEDICATORIA	vii
CONTENIDO GENERAL.....	viii
CONTENIDO DE CUADROS	x
CONTENIDO DE FIGURAS	xi
RESUMEN	xiii
PALABRAS CLAVE	xiii
ABSTRACT	xiv
KEY WORDS	xiv
CAPÍTULO I. ANTECEDENTES	1
1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. JUSTIFICACIÓN	3
1.3. OBJETIVOS	4
1.4. IDEA A DEFENDER	5
CAPÍTULO II. MARCO TEÓRICO	6
2.1. ESPAM MFL	6
2.2. CONTROL Y MONITOREO	8
2.3. INTERNET	8
2.4. LENGUAJES DE PROGRAMACIÓN	10
2.5. BASE DE DATOS	20

2.6. METODOLOGÍAS ÁGILES	21
CAPÍTULO III. DESARROLLO METODOLÓGICO	24
3.1. MÉTODOS.....	24
3.2. TÉCNICAS.....	50
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	51
4.1. RESULTADOS.....	51
4.2. DISCUSIÓN	72
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....	74
5.1. CONCLUSIONES	74
5.2. RECOMENDACIONES	75
BIBLIOGRAFÍA	76
ANEXOS.....	80

CONTENIDO DE CUADROS

2.1 Característica de los tipos de programación.....	18
3.1. Caso de uso 1/ CU_1, Ingreso o Login al Sistema.....	29
3.2. Caso de uso 1/ CU_2, Ingreso mediante mapas de uso.....	30
3.3. Caso de uso 1/ CU_3, Salir del sistema o Logout.....	30
3.4. Caso de uso 1/ CU_4, Labores Agronómicas.....	31
3.5. Caso de uso 2/ CU_5, Administrador.....	32
3.6. Caso de uso 3/ CU_6, Autenticación del usuario-Ingreso al sistema.....	34
3.7. Caso de uso 4/ CU_7, Generar Informes.....	35
3.8. Diccionario de datos de la tabla “Archivo”.....	38
3.9. Diccionario de datos de la tabla “Áreas Agrícolas”.....	38
3.10. Diccionario de datos de la tabla “Bodega”.....	38
3.11. Diccionario de datos de la tabla “Campos”.....	39
3.12. Diccionario de datos de la tabla “Datos Meteorológicos”.....	40
3.13. Diccionario de datos de la tabla “Imagen”.....	40
3.14. Diccionario de datos de la tabla “Labores”.....	40
3.15. Diccionario de datos de la tabla “Persona”.....	41
3.16. Diccionario de datos de la tabla “Producto”.....	41
3.17. Diccionario de datos de la tabla “Responsable”.....	41
3.18. Diccionario de datos de la tabla “Tipo”.....	41
3.19. Diccionario de datos de la tabla “Usuario”.....	42
3.20. Diccionario de datos de la tabla “transacciones”.....	42
4.1.Comparación de los procesos con el antes y después del sistema.....	67

CONTENIDO DE FIGURAS

3.1. Caso de uso de Coordinador del Área de Producción.....	28
3.2 Caso de uso Administrador.....	32
3.3. Caso de uso de Usuario.....	33
3.4. Caso de uso de Director de Carrera.....	35
3.5. Primer prototipo del sistema.....	36
3.6. Primer prototipo del sistema.....	37
3.7. Primer prototipo del sistema.....	37
3.8. Diagrama E/R de la base de datos.....	44
3.9. Captura de las tres capas usadas en el sistema.....	45
3.10. Modelo de base de datos en el DataSet de la Capa Datos.....	45
3.11. Diagrama de clases de la capa negocio.....	46
3.12. Vista del mapa realizado en Adobe Flash.....	47
3.13. Consulta en SQL Server para obtener las cosechas del área cacao clonal.....	48
3.14. Consulta en SQL Server para obtener la bodega del área cacao.....	48
3.15. Consulta en SQL Server para obtener datos de Personas.....	49
3.16. Ingreso de Persona por el usuario Administrador.....	49
3.17. Ingreso de Bodega por el usuario Administrador.....	49
4.1. Diagrama E/R de la base de datos.....	52
4.2. Inicio de sesión del sistema.....	54
4.3. Mapa de acceso a las áreas.....	55
4.4. Ventana principal del sistema con su respectivo menú.....	55
4.5. Pantalla donde se permite seleccionar la labor agronómica.....	56
4.6. Registro de labor agronómica.....	56
4.7. Registro de responsable	57
4.8. Registro de cultivos	57
4.9. Informes de áreas.....	58
4.10. Informe de cultivos.....	58
4.11. Cuadro comparativo del rendimiento del sistema.....	71

4.12. Vista del sistema AGROCON 3.0.....73

RESUMEN

Con el objetivo de mantener un efectivo registro de información de las áreas de producción de la Carrera de Agrícola se creó un Sistema de Control y Monitoreo. Para el efecto fue de vital importancia realizar un análisis a los métodos de registro de información y generación de reportes utilizados por los administradores. Mediante información obtenida gracias a entrevistas se detectó que estos contaban con pocos recursos que le permitieran optimizar el manejo de la misma, por lo que la implementación de un sistema les beneficiaría enormemente. Para la elaboración del sistema y demostrar que los resultados están respaldados técnicamente se trabajó con la metodología de desarrollo MIDAS, misma que consta de cuatro fases, dentro de las cuales se realizó la toma de requerimientos para definir los requisitos y la arquitectura de la aplicación web, diseñando un prototipo de la misma a fin de proporcionar al cliente una primera versión del producto. Continuando con el desarrollo de la base de datos y la programación de interfaces, que se desarrollaron en Microsoft SQL Server 2008 y en el entorno de desarrollo integrado de Microsoft Visual Studio en lenguaje C# y programación orientada a objetos y de esta manera verificar el correcto funcionamiento del sistema de control y monitoreo para su posterior implementación, finalizando el desarrollo con la fase de pruebas, que permitió evidenciar la reducción de un 92,98% en los tiempos en el ingreso de información y generación de reportes.

PALABRAS CLAVE

Áreas de producción agrícola, control y monitoreo, aplicación web.

ABSTRACT

In order to maintain effective information record production areas of the School of Agricultural System Monitoring and Control was created. For the effect was of vital importance to an analysis methods of recording and reporting information used by administrators. Using information obtained from interviews it was found that these had few resources to enable it to optimize the management of it, so the implementation of a beneficiary them greatly. To prepare the system and show that the results are supported technically worked with the development methodology MIDAS, it consists of four phases, in which the gathering of requirements was performed to define the requirements and architecture of the web application designing a prototype of it to provide the customer with a first version of the product. Continuing the development of database and programming interfaces, which were developed in Microsoft SQL Server 2008 and the integrated development environment for Microsoft Visual Studio C # language and object-oriented programming and thus verify proper operation the control and monitoring system for subsequent implementation, development finalizing the testing phase, which allowed demonstrate the reduction of 92,98% in the time data entry and report generation.

KEY WORDS

Areas of agricultural production, control and monitoring, web application.

CAPÍTULO I. ANTECEDENTES

1.1. PLANTEAMIENTO DEL PROBLEMA

Para Tello (2013) La agricultura es un sistema creado a partir de un ecosistema natural, manejado por la mano del hombre. Según De la Fuente *et al.* (2008) la agricultura se remonta a las épocas en que el hombre reemplazó sus hábitos nómadas por sedentarios. La agricultura ocupó y alteró progresivamente los espacios terrestres hasta cubrir una gran proporción de la superficie del planeta.

Manabí es una provincia especializada en agricultura, en relación con el resto del país. Además, este sector agrícola es uno de los más diversos del Ecuador, debido al tamaño de la provincia, la estabilidad climática y la topografía de sus suelos aptos para cultivos. Registra una alta especialización en los productos agrícolas: café, cacao, plátano, maíz duro seco, arroz, y frutas cítricas como limón, toronja, naranja, mandarina y maracuyá (Invest Manabí, s.f.)

Los habitantes del cantón Bolívar siempre se han identificado con la agricultura y la ganadería. Ello exigía en Calceta la presencia de un centro de estudios superiores en las áreas agrícola y pecuaria, de manera que la población estudiantil, pudiera profesionalizar los conocimientos en dichas áreas alcanzando un título académico, con la finalidad de servir no solo al cantón, sino a toda la región. No obstante la ESPAM MFL fue creada con el objetivo de atender estas demandas, es por esto que se crea la Carrera de Agrícola, la misma que cuenta con 12 áreas de producción que son, Cacao INIAP, Cacao Joven, Cacao Clonal, Cacao Injerto, Cacao Más Plátano, Área Convencional, Teca, Cultivos Transitorios, Frutales, Área Orgánica, Platanera y Vivero.

Los técnicos de las áreas de producción agrícola tienen pocos recursos para llevar un correcto control y monitoreo en cada una de ellas, esto impide tener un análisis concreto de lo que en realidad se usa para cada producción,

impidiendo fortalecer el cambio de la matriz productiva, lo que permitiría generar más riqueza, logrando un desarrollo más sostenible impulsando actividades ligadas al talento humano, a la tecnología y el conocimiento, a través del cambio de la matriz productiva. (SENPLADES, 2012).

Puesto que en el cambio de la matriz productiva, al igual que el conocimiento y el talento humano, el desarrollo de las nuevas tecnologías juega un papel importante dando respuesta a las exigencias de los procesos productivos en cuanto a rendimiento, calidad y eficiencia, las autoras del trabajo investigativo plantearon la siguiente interrogante:

¿Qué recurso tecnológico utilizar para llevar un correcto control y monitoreo en las áreas de producción agrícola de la ESPAM MFL?

1.2. JUSTIFICACIÓN

La sociedad moderna demanda alimentos variados y de buena calidad, esto implica un reto para los agricultores de frutas, hortalizas y granos, ya que deben producir lo suficiente para obtener ganancias y al mismo tiempo, evitar la contaminación del producto requerido por la población, asegurando una producción exitosa.

Las autoras del trabajo de investigación desarrollaron un sistema de control y monitoreo en las áreas de producción agrícola de la ESPAM MFL, a fin de realizar verificaciones constantes y comprobar que la implementación avanza como se planificó.

En el Ecuador, la Agencia Ecuatoriana de Aseguramiento de la Calidad del Agro, propone el control y monitoreo en plantas y vegetales para mantener y/o mejorar el estatus fitosanitario del país mediante el conocimiento, la prevención de ingreso y apoyo al manejo de plagas, así como contribuir a la producción de plantas y productos vegetales en condiciones fitosanitarias, según las exigencias del comercio nacional e internacional (Agrocalidad, 2008).

Además, El Plan Nacional para el Buen Vivir en su objetivo número 10 plantea la transformación de la matriz productiva, impulsando un modelo productivo más democrático, incluyente, sostenible y responsable, fundamentado en el conocimiento y talento humano que son recursos infinitos, obteniendo como resultado que el sector agrícola mejore la productividad en el campo, impulsando exportaciones y otorgando valor agregado, siendo menos vulnerables a los cambios en la economía internacional (SENPLADES, 2013).

Finalmente este trabajo de titulación se basa en el Art. 8 de la Ley Orgánica de Educación Superior (LOES, 2010) literal f indica: “Fomentar y ejecutar programas de investigación de carácter científico, tecnológico y pedagógico que coadyuven al mejoramiento y protección del ambiente y promuevan el desarrollo sustentable nacional”.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar un Sistema Web de Control y Monitoreo en las Áreas de Producción Agrícola de la ESPAM MFL, para mantener un efectivo registro de los procesos.

1.3.2. OBJETIVOS ESPECÍFICOS

- Definir los requisitos y la arquitectura de la aplicación web para establecer sus funcionalidades.
- Diseñar un prototipo de la página web para proporcionar al cliente una primera versión del producto.
- Desarrollar la base de datos y la aplicación web de las áreas de producción agrícola.
- Verificar el correcto funcionamiento e implementar el sistema.

1.4. IDEA A DEFENDER

La implementación de un sistema informático para el control y monitoreo mejorará el registro de los procesos permitiendo contar con reportes agilizados en las distintas áreas de producción que tiene la Carrera de Agrícola de la ESPAM MFL.

CAPÍTULO II. MARCO TEÓRICO

2.1. ESPAM MFL

2.1.1. HISTORIA DE LA ESPAM MFL

En la página web institucional ESPAM MFL (2015) se puede encontrar que Manabí es una provincia rica en variados recursos. Los contrastes se marcan con fuerza en sus 22 cantones: el mar y sus montañas; la cultura ancestral de su población chola y montubia, que sorprende con una y mil leyendas; su comida típica, muy apreciada por nativos y extraños; su artesanía.

Mas, los habitantes del cantón Bolívar han dirigido su mirada a la tierra, pródiga desde siempre y, en ese contexto, se han identificado con la agricultura y la ganadería. Vale recordar que, hubo épocas en que este cantón fue productor y exportador de caucho, madera de balsa, tagua, cacao y algodón; producción disminuida en las últimas décadas, por causas conocidas por todos; pero hoy, con la Presa La Esperanza y el Proyecto Carrizal-Chone, hombres y mujeres con renovados bríos, fincan, otra vez, su ilusión en la tierra.

Las gestiones, un largo recorrido, empezaron en el Congreso Nacional y luego en otras instancias desde 1995. Se crea así el INSTITUTO TECNOLÓGICO SUPERIOR AGROPECUARIO DE MANABÍ, ITSAM, mediante Ley N°. 116, publicada en el R.O. N°. 935, el 29 de abril de 1996.

Tres años después, el Congreso Nacional expidió la Ley Reformatoria que transformaba el Instituto Tecnológico Superior Agropecuario de Manabí, ITSAM, en ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ, ESPAM, cuya Ley 99-25 fue publicada en el R.O. el 30 de abril de 1999.

La Escuela Superior Politécnica Agropecuaria de Manabí nace como persona jurídica de derecho público, autónoma, que se rige por la Constitución Política del Estado, Ley de Educación Superior, su Estatuto Orgánico y Reglamentos, para preparar a la juventud ecuatoriana y convertirla en profesionales, conforme lo exigen los recursos naturales de su entorno. La ESPAM inicia sus labores con las carreras de Agroindustria, Medio Ambiente, Agrícola y Pecuaria. Posteriormente, mediante un estudio de mercado, se crea la carrera de Informática, emprendiendo así, un riguroso programa de fortalecimiento académico, con el fin de formar profesionales idóneos que ejecuten proyectos sustentables, generadores de fuentes de trabajo.

Ante la demanda de nuevas carreras, los directivos de la ESPAM, no han escatimado esfuerzos para incrementar otras, de tipo empresarial. Es así como desde el año 2003 funcionan dos nuevos programas: Administración Pública y Administración de Empresas, los que se cumplen en horarios nocturnos, al igual que la Carrera de Informática. A partir del año 2007 y, producto de un estudio, los estudiantes tienen una nueva opción: Ingeniería en Turismo. Con ello se busca potenciar a la población manabita, ávida de lograr una profesión acorde con sus aspiraciones.

La Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López tiene dos áreas que son la Agroindustrial y Agropecuaria, dentro de ellas existen 29 áreas. Estas son creadas para que los estudiantes pongan en práctica sus conocimientos, la carrera de agrícola tiene áreas llamadas:

- Cacao
- Frutales
- Cultivo Convencional
- Orgánica

2.2. CONTROL Y MONITOREO

2.2.1. CONTROL

Franco (2011) menciona a Henry Fayol al decir que el control consiste en verificar si todo ocurre de conformidad con el plan adoptado, con las instrucciones emitidas y con los principios establecidos. Tiene como fin señalar las debilidades y errores a fin de rectificarlos e impedir que se produzcan nuevamente.

2.2.2. MONITOREO

Arana (2011) define al monitoreo como un proceso continuo y sistemático en el cual se verifica la eficiencia y la eficacia de un proyecto o programa. Permite identificar logros y debilidades y en consecuencia, recomienda medidas correctivas para optimizar los resultados esperados.

Existen diversas maneras de realizar un sistema de control y monitoreo, sea con una aplicación de escritorio o una aplicación web.

2.3. INTERNET

Según Dominguez *et al.* (2012) con el término de Internet se conoce al sistema de comunicación mundial compuesto por redes de computadoras enlazadas entre sí, posibilitando el acceso fácil a grandes fuentes de información. Para utilizar los servicios de Internet además de las computadoras se necesita de un modem y un programa navegador así como establecer la conexión con un servidor mediante la línea telefónica.

Gran parte de la información en Internet se presenta en el formato de páginas web. Las páginas destinadas a organizar la información se denominan portales. Las páginas destinadas a encontrar la información mediante palabras clave se denominan buscadores. La información disponible en Internet hay que saberla

seleccionar, sistematizar e interpretar, esto ha motivado el incremento llamativo de su uso en toda la población. Internet ha acortado las distancias y ha facilitado el flujo de información, llegando a formar parte de la vida diaria de muchas personas.

Puede afirmarse que Internet está basada en unas pocas características técnicas innovadoras, de las cuales las más importantes son los protocolos TCP/IP -el software que permite la comunicación entre ordenadores con configuraciones diferentes y que establece qué tipo de formatos debe tener la información para poder viajar de uno a otro.

La expansión de Internet es, sin duda, una de sus características más remarcable: su tasa de crecimiento es muy superior a la de cualquier otro medio de comunicación anterior y sólo ha sido superada por la expansión de la telefonía móvil. El internet es utilizado en todos los campos de la ciencia, particularmente se afirma que el uso intensivo de Internet por parte de las empresas genera el llamado e-business, el comercio y el marketing electrónicos, e-commerce; el uso de la red para las relaciones personales genera la denominada sociabilidad virtual o las llamadas comunidades virtuales; el uso por parte de gobiernos y administraciones produce el e-government y la administración electrónica (Aibar, 2008).

2.3.1. APLICACIÓN WEB

La web se ha convertido en un enorme repositorio de información textual semi-estructurada que abarca casi todas las áreas del conocimiento humano. La literatura universal, el conocimiento científico y la prensa digital son algunos de los contenidos web que consumen a diario millones de personas alrededor del mundo. La web es ahora mucho más social. En el 2010 fueron creados aproximadamente 152 millones de blogs en Internet, registradas 175 millones de personas en Twitter y 600 millones en Facebook (Hidalgo y Rodriguez, 2013).

Por otra parte Rodríguez *et al.* (2013) hace énfasis al decir que las aplicaciones web son convenientes para acceder desde cualquier lugar usando Internet. En vez de crear clientes para diferentes plataformas, la aplicación web se escribe una vez y se ejecuta igual en todas partes.

2.3.2. TIPOS DE APLICACIÓN WEB

Oliveros *et al.* (2011) muestra los siguientes tipos de aplicaciones web:

- Sitios Web centrados en documentos
- Aplicaciones Web Interactivas
- Aplicaciones Web transaccionales
- Aplicaciones Web basadas en flujos de trabajos
- Aplicaciones Web de colaboración
- Web social.
- Aplicaciones Web orientadas a portales.
- Aplicaciones de Web ubicua.
- Web semántica.

2.4. LENGUAJES DE PROGRAMACIÓN

Para Buitrago (2010) el Lenguaje de Programación es el medio de comunicación entre un programador y una computadora; es mediante el lenguaje de programación que el programador le dice a la computadora qué acciones debe ejecutar, para resolver un problema computacional.

Por otra parte, Zapata y Chaverra (2010) ha señalado que los lenguajes de programación son un sistema para describir conceptos en una forma legible para la máquina. Un lenguaje de programación es el fundamento para el código que se genera.

Para López *et al.* (2009) en la actualidad existen infinidad de lenguajes de programación con diversidad de aplicaciones tales como inteligencia artificial,

tutoriales inteligentes, programación multimedia, entre otros. Así mismo, menciona que un lenguaje de programación está formado por un conjunto de símbolos básicos, alfabeto, y de reglas que especifican como manipularlos; además de darle significado a las cadenas formadas al manipular los símbolos básicos.

2.4.1. LENGUAJE DE PROGRAMACIÓN PARA LA WEB

Cáceres y Pinto, (2011) mencionan que en la actualidad el desarrollo de sistemas de información en Plataformas Web no sólo es un hecho, sino que está reemplazando las tradicionales aplicaciones Desktop. En cierta forma todo tiende a la Web a que nuestras aplicaciones puedan ser accedidas desde cualquier lugar del mundo, sin necesidad de instalar software, sin limitaciones en la usabilidad y sin que nuestro hardware tenga que sufrir cambios por requerimientos del sistema en sí.

Para Jabba *et al.* (2004) la humanidad ha sido testigo de la forma en que los websites pioneros han pasado a ser maravillosos sitios interactivos de hoy en día, y en este punto existe la necesidad de saber elegir cual herramienta utilizar. Por ejemplo, ASP provee el ambiente de desarrollo más fácil, mientras que JSP requiere el conocimiento de Java y Servlets. PHP es una tecnología popular basada en una fuente de desarrollo abierta. ASP se acomoda mejor para sitios pequeños que tengan entre 100-500 consultas por día, mientras que JSP es capaz de manejar entre 100 y un millón de consultas por día.

Puesto que la programación de sitios Web ha generado la creación de varias herramientas de desarrollo, es importante identificar cuáles ofrecen un mejor rendimiento, por esta razón se optó por usar la herramienta de Visual Studio .Net.

2.3.2. CALIDAD DE LOS SITIOS WEB

Para Gomes y De Faria (2012) la calidad en los servicios en Internet es de particular relevancia para el desarrollo y la administración de los sitios. La evaluación de los sitios ha evolucionado con el crecimiento y la maduración de Internet, pasando inicialmente de los análisis sobre aspectos puramente técnicos a la obtención de información sobre las expectativas y las percepciones de los usuarios.

El mismo autor cita a Lohse y Spiller (1998) las principales métricas usualmente utilizadas por las empresas presentes en Internet son: cantidad de usuarios, cantidad de visitas, visualización de las páginas, ítem, y tasa de conversión división del número de usuarios que ejecutaron una determinada tarea por el número total de usuarios que accedieron al sitio.

Una actividad muy común es hacer pruebas de software, para Fernández (2005) las pruebas de software son seguramente la actividad más común de control de calidad realizada en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas. Las pruebas de software se definen como una actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto.

Para Moreno *et al.* (2008) el objetivo principal de la calidad de un software es proporcionar al usuario final de un producto de software "una garantía de los requisitos necesarios para su óptima utilización" específicamente en aplicaciones web. Para poder realizar una aplicación web se necesita hacerlo en un lenguaje de programación.

2.4.1.1. VISUAL STUDIO .NET

Según Salazar *et al.* (2012) Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de

programación como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic.NET.

El mismo autor afirma que a través de la plataforma .NET es posible interactuar con otros componentes realizados en otros lenguajes .NET de manera muy fácil, también se puede interactuar con componentes no gestionados fuera de la plataforma .NET, por esta razón se puede integrar con facilidad en sistemas ya creados, al poseer una librería de clases muy completa y muy bien diseñada, esta nos permitirá disminuir en un gran porcentaje los tiempos de desarrollo.

Una de las herramientas que ofrece Visual Studio .Net para el desarrollo de aplicaciones web es ASP .NET.

2.4.1.2. ASP .NET

Según Moreano (2014) ASP.NET es una herramienta que permite el desarrollo de páginas web dinámicas, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

ASP.NET incluye ventajas como: instalación más fácil; biblioteca Modernizer, una interesante herramienta que facilita la detección de características de browsers que permite el uso de tecnologías modernas como HyperText Markup Language 5 (HTML5) o Cascading Style Sheets 3 (CSS3), mientras que se mantiene la compatibilidad con navegadores anteriores (Martinez *et al.* 2014). Zapata y Sierra (2005) dicen que ASP® (Active Server Pages) es una tecnología en la cual es posible desarrollar aplicaciones ejecutables desarrolladas bajo VBScript y JavaScript que corren bajo un servidor Web y que permiten tener acceso programático a los recursos de éste, para poder realizar un amplio conjunto de funciones; además, ASP® brinda la posibilidad de realizar cálculos y ejecutar secciones completas de código de una página antes de que ésta sea enviada al cliente. El gran inconveniente de esta

tecnología radica en el hecho de que las herramientas para la depuración del código brindadas, no siempre funcionan de la forma esperada.

Los mismos autores explican que ASP.NET® es una evolución del ASP® y se constituye en un ambiente de programación construido sobre el entorno NGWS (New Generation Windows Services), que permite crear poderosas aplicaciones de Internet y Servicios Web que corren sobre IIS (Internet Information Server). ASP.NET® no sólo sirve para la creación de aplicaciones Web (que bien podrían ser creadas usando otras tecnologías como CGI); además de esta funcionalidad, se pueden resaltar en ASP.NET® sus capacidades de integración con el servidor Microsoft, que posibilita el acceso a datos y el control de herramientas de seguridad. Una aplicación ASP.NET® incluye las siguientes partes:

- Web Forms (Formularios tipo Web) o páginas aspx.
- Código asociado con el Web Form, el cual queda al lado del servidor.
- Los archivos de configuración, los cuales son archivos XML, que definen las características predeterminadas de la aplicación y el servidor Web.
- El archivo Global.asax, el cual contiene el código necesario para los eventos que se dan a nivel de la aplicación cuando es iniciado el ASP.NET®.
- Enlaces a XML Web Service que permiten que la aplicación envíe y reciba datos desde un XML Web Service.
- Conectividad a bases de datos.
- Sistema de caché, que permite que la aplicación retorne más rápidamente una Web Form.

Una herramienta que se utilizó es Crystal Report, para Stangarlin *et al.* (2013) es una herramienta estándar para la creación e integración de reportes con datos provenientes de múltiples fuentes de datos. Permite transformar rápidamente cualquier fuente de datos en contenido interactivo con calidad de presentación en la plataforma .NET lo que ha supuesto una ventaja fundamental durante años; Además los usuarios finales pueden acceder e

interactuar con los reportes a través de portales web, dispositivos móviles y documentos de Microsoft Office.

ASP .NET se lo va a usar con el lenguaje C#, Bastarache *et al.* (2012) define a C# como un lenguaje de programación diseñado por Microsoft en 2001 como parte de su plataforma .NET. Combina el lenguaje de bajo nivel de C y la velocidad de la programación de alto nivel de Visual Basic. C# es un lenguaje orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, es simple pero eficaz y está diseñado para escribir aplicaciones empresariales.

2.4.2. TIPOS DE PROGRAMACIÓN

Existen diversos tipos de programación, Carranza y Ganchozo (2013) citan a García (2010) expresa que los tipos o técnicas de programación son bastantes variados, la idea es que el programador los conozca y sepa seleccionar el adecuado para resolver el problema que se le plantea, algunos tipos de programación son:

- Programación estructurada (PE)
- Programación orientada a eventos
- Programación orientada a objetos (POO)
- Programación lógica
- Programación en capas
- Programación modular

A continuación se muestra un cuadro comparativo de los tipos de programación:

Tipo de programación

Programación estructurada (PE)

La metodología de programación estructurada se fundamenta en técnicas de segmentación, la cual plantea que un problema se puede dividir en problemas más pequeños (módulos) y más simples de resolver, de tal forma que la suma de las soluciones de cada problema sea el resultado de la solución total de éste. Este hecho es importante debido a que es mucho más fácil comprender completamente el trabajo que realiza una función determinada si todas las instrucciones que influyen en su acción están físicamente contiguas y encerradas por un bloque. Posibilitando a la vez en mayor grado la reutilización del código en futuras aplicaciones. Sin embargo al obtener un único bloque del programa, cuando este se hace demasiado grande puede ser problemático para el manejo del código fuente.

Programación orientada a eventos

Este es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen. El creador de un programa dirigido por eventos debe definir los eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos, lo que se conoce como el administrador del evento. Pese a estas ventajas este tipo de programación supone una complicación añadida con respecto a otros paradigmas de programación, debido a que el flujo de ejecución del software escapa al control del programador, puesto que en la programación clásica el flujo estaba en poder del programador y era este quién decidía el orden de ejecución de los procesos, mientras que en programación orientada a eventos, es el usuario el que controla el flujo y decide.

Programación orientada a objetos (POO)

La programación orientada a objetos surge en la historia como un intento para dominar la complejidad que, de forma innata, posee el software, puesto que, Al dividir el problema en partes más pequeñas se puede probarlas de manera

independiente y aislar mucho más fácilmente los posibles errores que puedan surgir. En este tipo de programación encontramos grandes ventajas como los programas son más sencillos de leer y comprender, teniendo la facilidad de añadir, suprimir o modificar nuevos objetos que permite hacer modificaciones de una forma muy sencilla y su utilización en distintas partes del programa y en numerosos proyectos.

Programación lógica

La programación lógica, forma parte de lo que se conoce como programación declarativa. En los lenguajes tradicionales, la programación consiste en indicar cómo resolver un problema mediante sentencias; en la programación lógica, se trabaja de una forma descriptiva, estableciendo relaciones entre entidades, indicando no cómo, sino qué hacer. La programación lógica consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación. Encontrando su hábitat natural en aplicaciones de inteligencia artificial

Programación en capas

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles siguiendo el modelo, el desarrollador se asegura avanzar en la programación del proyecto de una forma ordenada, lo cual beneficia en cuanto a reducción de costos por tiempo, debido a que se podrá avanzar de manera más segura en el desarrollo, al ser dividida la aplicación general en varios módulos y capas que pueden ser tratados de manera independiente y hasta en forma paralela.

Programación modular

La meta de la programación modular es obtener una organización de la arquitectura del software, con un diseño no orientado a la simple obtención del programa final, sino también a su mantenimiento, a su reutilización y a poder probarlo y entenderlo de forma fácil. La programación modular consta de varias secciones divididas de forma que interactúan a través de llamadas a procedimientos, que integran el programa en su totalidad. En la programación modular, el programa principal coordina las llamadas a los módulos secundarios y pasa los datos necesarios en forma de parámetros. A su vez cada módulo puede contener sus propios datos y llamar a otros módulos o funciones.

Cuadro 2.1: Característica de los tipos de programación

Fuente: Las autoras.

Una vez analizados los diversos tipos de programación, las autoras deciden que los tipos de programación a utilizar para realizar la aplicación web es programación orientada a objetos, puesto que este tipo de programación facilita la división del problema en partes más pequeñas pudiendo probarlas de manera independiente y facilitando la añadidura, supresión o modificación de nuevos objetos. Así mismo se empleará la programación por capas ya que esta cuenta con una gran ventaja, permite desarrollar la aplicación en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado, facilitando también la distribución del trabajo por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles siguiendo el modelo, beneficiando en cuanto a reducción de costos por tiempo y mejoras a la aplicación sin que esto sea una tarea tediosa y desgastante.

2.4.2.1. PROGRAMACIÓN ORIENTADA A OBJETOS

Para Rob y Coronel (2004) la programación orientada a objetos (POO), fue desarrollada como métodos alternativos de métodos de programación tradicionales. En un ambiente POO, el programador crea o utiliza objetos,

módulos reutilizables, autónomos que contienen datos, así como los procedimientos utilizados para operar con dichos datos.

Los mismos autores afirman que los conceptos Orientado a Objeto aparecieron en lenguajes de programación como Ada, ALGOL, LISP y SIMULA. Estos lenguajes de programación dieron pie a la introducción de conceptos OO más refinados. Smalltalk, C++ y Java son lenguajes de programación orientados a objetos (LPOO). Java se utiliza para crear aplicaciones Web que funcionan en el internet y son independientes de los sistemas operativos.

Stair y Reynolds (1999) señalan que los lenguajes de programación orientados a objetos ofrecen las ventajas potenciales de los códigos reutilizables, costos inferiores, menos pruebas y rapidez en la puesta en marcha. En lugar de escribir líneas de código detalladas, los programadores pueden combinar, modificar e integrar módulos predesarrollados en un programa unificado. Incluso con modificaciones, el uso del método modular puede ser mucho más rápido que escribir todos los enunciados desde cero. Las desventajas potenciales de los programas orientados a objetos incluyen tiempos de ejecución más lentos y necesidades de memoria más altas. Sin embargo, ante el decremento en los costos del hardware y aumento en los sueldos de los programadores, muchos gerentes están dispuestos a confrontar estas desventajas a cambio de un tiempo de desarrollo de programas más rápidos.

2.4.2.2. PROGRAMACIÓN POR CAPAS

Álvarez y Montes *et al.* (2010) Expresan que la programación en capas, no es nada más que un estilo de programación donde su objetivo primordial es separar diferentes aspectos de desarrollo, separando la lógica de negocio de la lógica de diseño; para luego unirlos ya sea en código o en tiempo de ejecución, de aquí salen la programación en dos capas, tres capas, cuatro capas, n-capas. Para el desarrollo de esta aplicación se utilizará la programación en tres capas.

Para Santos (2006) una aplicación de tres capas es un tipo de aplicación donde existe una tercera capa (de aplicación) entre las dos utilizadas tradicionalmente (cliente y servidor). Esta tercera capa se encarga de separar el procesamiento de la visualización. Una de las principales ventajas de esta filosofía de programación radica en la utilización de determinadas reglas para todas las aplicaciones que las utilizan, evitando la pesada tarea de tener que duplicar reglas para cada aplicación mientras que se aprovechan las capacidades de los servidores (máquinas por definición más potentes que aquellas en que se ejecutan las aplicaciones).

Para poder realizar un sistema, sea este WIN32 o una aplicación web se deben realizar base de datos.

2.5. BASE DE DATOS

Rivera y Zambrano (2012) citan a Enríquez *et al.* (2011) para definir a las base de datos como un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

Asimismo indica que en informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental. Uno de estos gestores de base de datos es Microsoft SQL Server.

2.5.1. MICROSOFT SQL SERVER

Salazar *et al.* (2012) define a Microsoft SQL Server como un sistema para la gestión de base de datos creado por Microsoft basado en el modelo relacional, sus lenguajes de consultas son T-SQL y ANSI SQL. Entre los beneficios que brinda el motor de base de datos se encuentra que tiene compatibilidad integrada con XML, cuenta con características de escalabilidad, disponibilidad y seguridad, las necesarias para manejar los componentes de almacenamiento de sitios Web de gran tamaño.

El mismo autor señala que SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, así como para configurarlos y administrarlos. SQL Server Management Studio combina un amplio grupo de herramientas gráficas con un editor de texto enriquecido para ofrecer acceso a SQL Server a los programadores y administradores, sin importar su nivel de especialización.

Zapata y Sierra (2005) mencionan que las Herramientas CASE o Computer-Aided Software Engineering, son un conjunto de aplicaciones informáticas que dan asistencia a los analistas y desarrolladores, durante todo el ciclo de vida del software. Estas herramientas se destinan a aumentar la productividad y reducir costos en tiempo y dinero. Son importantes porque muchas de ellas contienen como una opción la generación de código, aunque en la actualidad lo hacen de forma incompleta.

Se debe tomar en cuenta una metodología al momento de desarrollar un software, las más utilizada para el desarrollo de páginas web son las denominadas metodologías ágiles.

2.6. METODOLOGÍAS ÁGILES

Según Rodríguez *et al.* (2010) Agile Software Development (ADS) define un grupo de metodologías de desarrollo software basadas en un desarrollo

iterativo, donde los requisitos y las soluciones evolucionan a través de una estrecha colaboración entre equipos inter-funcionales y auto-organizados.

Las metodologías ágiles contemplan el desarrollo de software de manera integral, con un énfasis especial en la entrega de valor al cliente, en la generación de negocio y el retorno de la inversión (ROI). Sólo hay una manera efectiva de crear software que funcione, y es de manera colaborativa. La colaboración entre cliente y desarrolladores es indispensable: se debe fomentar y apoyar (Díaz, J. 2009).

2.6.1. METODOLOGÍA MIDAS

La metodología MIDAS es muy usada, Zorrilla *et al.* (2007) define a MIDAS como 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 del aspecto del sistema a modelar. 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.

Para el desarrollo de la metodología se usaron los diagramas de casos de uso, para Vidal *et al.* (2012) dichos diagramas son una técnica de Ingeniería de Software para dirigir completamente el ciclo de vida del desarrollo de software, donde es relevante la identificación de Incumbencias para su modelación, implementación y validación. Justamente, los diagramas de casos de uso permiten expresar Incumbencias de usuario, y proveer una temprana validación de las mismas.

Zapata y Tamayo (2009) mencionan que en la especificación de la Superestructura del Unified Modeling Language UML, el diagrama de casos de uso se define como el “diagrama que muestra las relaciones entre los actores y el sujeto (sistema) y los casos de uso, describe los requisitos funcionales del

sistema en términos de las secuencias de acciones. Se presentan los siguientes elementos de su especificación:

- Casos de uso: son las especificaciones de un conjunto de acciones realizadas por el actor sobre el sistema.
- Actores: son los roles que los usuarios desempeñan respecto del sistema y que emplean los casos de uso.
- Relaciones: identifican la comunicación existente entre actores y casos de uso.

Las principales ventajas de utilizar este diagrama, son:

- La captura de los requisitos funcionales desde el punto de vista del usuario.
- La utilización de los casos de uso para educir y documentar los requisitos funcionales.
- El manejo de la complejidad en sistemas robustos, descomponiendo el problema en funciones más simples

CAPÍTULO III. DESARROLLO METODOLÓGICO

3.1. MÉTODOS

3.1.1. METODOLOGÍA MIDAS

MIDAS tiene una arquitectura por modelos, Marcos *et al.* (2004) afirma que esta posee un núcleo central que representa los modelos de dominio y de negocio. Sobre dicho núcleo, posee un anillo que representa el modelado de la dimensión estructural y de comportamiento del sistema. En la capa externa se sitúa la plataforma o tecnología concreta soportada. MIDAS selecciona, adapta e integra, siempre que sea posible, las técnicas y notaciones de otras metodologías y define otras nuevas para aquellos casos en que sea necesario. Así, por ejemplo, para el modelado de la dimensión estructural, se ha definido una extensión UML para el diseño de BD objeto-relacionales y también una extensión para representar XML Schemas.

Rivera y Zambrano (2012) citan a Vela (2003) para definir las fases de la metodología MIDAS, estas son:

3.1.1.1. MIDAS/SD

Es la primera iteración, que constituye el núcleo del proceso, se definen los requisitos y la arquitectura del sistema. Asimismo la fase de análisis, una arquitectura del software independiente a la funcionalidad que permite la combinación de diferentes modelos de arquitectura con diferentes modelos funcionales. Considerar la arquitectura del software desde las fases iniciales del desarrollo del SIW permite dirigir su desarrollo y determinar la capacidad para evolucionar el sistema. Además, provee de un mecanismo de reutilización mediante el uso de patrones de arquitectura como respuesta de los requisitos no funcionales definidos por el usuario.

Para cumplir con lo anteriormente mencionado se definieron los requisitos y la arquitectura del sistema y así ser eficaces al momento de realizarlo. Se dio prioridad a las necesidades de las áreas de producción Agrícola, para determinar la fase de inicio del desarrollo; para lo cual se efectuó una entrevista informal al Director de la Carrera de Agrícola, como se muestra en el anexo 1 y anexo 2.

Tomando en cuenta las actas 1, 3, 4 y 5 que se muestran en el anexo 7 se determinaron los requerimientos de acuerdo al análisis de la información:

Resoluciones del acta 1:

- Acceder al sistema por medio de Login
- Ingreso mediante mapas de uso
- Registro de la producción por cultivos2
- Registro de la producción por épocas del año
- Registro estadístico de información
- Ingreso de la planificación por cultivos
- Ingreso de la planificación por época del año
- Control de la planificación de los cultivos
- Control de la planificación de cada época del año
- Alertas de corrección por cada cultivo
- Alertas de corrección por época del año
- Registro de actividades por cada cultivo
- Reportes semanales por épocas del año y cultivos
- Reportes mensuales por épocas del año y cultivos
- Reportes estadísticos semanales por épocas del año y cultivos
- Reportes estadísticos mensuales por épocas del año y cultivos
- Reportes de actividades por cada cultivo
- Reportes de alertas de corrección por épocas del año y cultivos

Resoluciones del acta 3:

Se determinó que el Sistema ya no tendrá ingresos por Cultivos y Épocas del año, se lo hará mediante un menú de Labores Agronómicas, este menú tendrá las siguientes opciones:

- Siembra
- Fertilización
- Labores Fitosanitarias
- Cosechas
- Pos-Cosechas
- Otros
- Informes

Cabe recalcar que la opción Informes solo estará habilitada para el Director de Carrera. Así mismo se generaran para cada uno de los menús, es decir, para siembra, fertilización, y uno general que abarque todas las opciones.

También se definieron los campos a ingresarse en cada uno de los menús, estos son:

- Fecha
- Nombre de material
- Cantidad de material
- Unidad de medida del material
- Personal
- Responsable
- Valores Totales

Todas las áreas de producción tendrán las mismas opciones de ingreso, y cada coordinador solo tendrá habilitada su área respectiva.

Resoluciones del acta 4:

Se determinó que el Sistema tendrá ingreso de Datos Meteorológicos, este se lo realizara diariamente, para poder visualizarlo se lo hará en una pestaña con el Menú de Datos Meteorológicos. Así mismo, se ingresara un Calendario con las fases lunares del año, esto para saber las fechas de poda, de siembra, entre otras.

Se agregara un Menú con el nombre de Bodega, aquí se realizara la consulta de los equipos y productos que existen, y tendrá la opción para nuevos ingresos. De los productos y equipos se podrá ingresar y consultar los campos nombre, cantidad, precio y observaciones, es decir, será especie de inventario.

Resoluciones del acta 5:

Se determinó que el Sistema solo tendrá 6 tipos de usuarios para las áreas agrícolas, estos son usuario Cacao, que abarca las áreas de cacao clonal, cacao joven, cacao injerto, cacao INIAP y cacao más plátano, Convencional, Orgánico, Frutales y Plátano para cada una de las áreas correspondientes, y usuario Administrador de Campo para las áreas restantes.

Así mismo se creara un usuario Meteorológico, este tendrá acceso al ingreso del calendario lunar, datos meteorológicos y algún boletín de un evento o predicciones. Los usuarios Administrador y Director quedan como ya se había establecido anteriormente.

Las áreas de Cacao, Convencional, Orgánico, Frutales y Plátano tendrán una bodega cada una, así mismo podrán revisar la bodega general para poder solicitar algún producto a esta, mientras que las áreas que correspondan al usuario Administrador de Campo tendrán únicamente acceso a la bodega principal.

Luego de haber cumplido con la primera fase de esta metodología se llevó a cabo el desarrollo de la segunda fase de la metodología.

3.1.1.2. MIDAS/HT

En la segunda iteración, se desarrolla un primer prototipo del SIW, construyendo el hipertexto con páginas estáticas en HTML para proporcionar al cliente una primera versión del producto en un corto periodo de tiempo. Para cada actividad de ésta iteración se definen una serie de tareas, técnicas y notaciones.

Para desarrollar el primer prototipo de la aplicación, se realizaron los casos de uso correspondientes al análisis de la información que se realizó en el objetivo anterior. Quedando así determinadas las siguientes funciones:

CASO DE USO 1

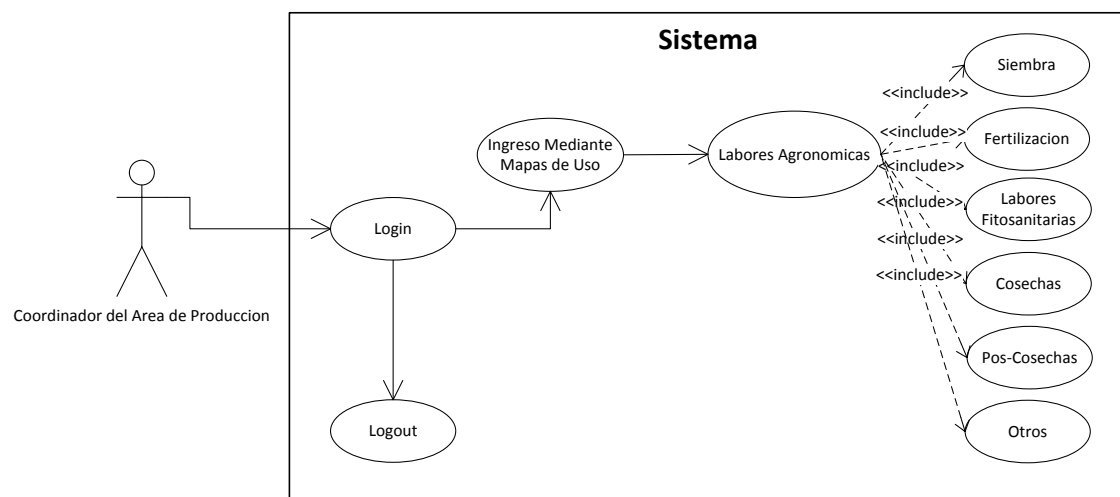


Figura 3.1. Caso de uso de Coordinador del Área de Producción
Fuente: Las Autoras

Cuadro 3.1. Caso de uso 1/ CU_1, Ingreso o Login al Sistema.

Nombre:	Ingreso o Login al Sistema / CU-1	
Actor(es):	Director de Carrera Agrícola, Coordinador de áreas de producción	
Descripción:	Ingreso al sistema web	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Estar en la sección de ingreso al sistema web	
	2. Completar el formulario de ingreso.	2. Muestra pantalla el formulario de ingreso.
	3. Enviar formulario de ingreso	3. Validación de datos, si los datos fueron correctamente registrados, crear cookies, sesión y temporales y redirigir al sistema, en caso de no ser así volver al formulario de ingreso mostrando un mensaje de ingreso incorrecto solicitando nuevos datos.
	4. Autenticación al Sistema.	4. Muestra pantalla del sistema
Precondición:	El usuario debe contar con un nombre de usuario y una contraseña definidos previamente para poder acceder a las opciones pertinentes en el sistema web.	
Poscondición:	El usuario accederá al sistema y tendrá privilegios específicos dependiendo del tipo de usuario	
Presunción:	El sitio web y la base de datos están disponibles.	

Cuadro 3.2. Caso de uso 1/ CU_2, Ingreso mediante mapas de uso

Nombre:	Ingreso mediante mapas de uso / CU-2	
Actor(es):	Director de Carrera Agrícola, Coordinador de áreas de producción	
Descripción:	Describe el proceso de ingresar mediante el mapa de uso de las áreas de producción agrícola	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Estar en la sección de registro en el sistema web	
	2. Seleccionar en el mapa de uso el área de producción en el que desea ingresar	2. Muestra un mapa topográfico de las áreas de producción agrícola de la ESPAM MFL
	3. Muestra formulario de tipo de registro	3. Muestra si se va a realizar el registro por siembra, cosecha, post cosecha, labores fitosanitarias, otros.
Precondición:	El usuario debe estar autenticado en el sistema.	
Poscondición:	El usuario se encuentra registrado. Y está disponible para el ingreso de la producción.	
Presunción:	El sitio web y la base de datos están disponibles.	

Cuadro 3.3. Caso de uso 1/ CU_3, Salir del sistema o Logout.

Nombre:	Salir del sistema o Logout/ CU-3	
Actor(es):	Director de Carrera Agrícola, Coordinador de áreas de producción	
Descripción:	Finaliza sesión del sistema	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Estar autenticado	

	en el sistema	
	2. Estar en la sección de Salir del sistema.	2. Muestra pantalla la sección de salir del sistema
	3. Seleccionar el botón de salir	3. Finaliza sesión, destruye cookies, sesiones y redirección a la página principal del sitio web
Precondición:	El usuario debe estar autenticado en el sistema, presionar el botón salir del sistema	
Poscondición:		
Presunción:	El sitio web y la base de datos están disponibles.	

Cuadro 3.4. Caso de uso 1/ CU_4, Labores Agronómicas

Nombre:	Labores Agronómicas/ CU-4	
Actor(es):	Director de Carrera Agrícola, Coordinador de áreas de producción	
Descripción:	Permite ingresar labores agronómicas	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Estar autenticado en el sistema	
	2. Haber ingresado a través de los mapas de uso	2. Muestra pantalla con los tipos de ingreso de la producción
	3. Seleccionar si se va a ingresar siembra, fertilización, labores fitosanitarias, cosechas, pos-cosechas y otros.	3. Envía a formulario correspondiente
Precondición:	El usuario debe estar autenticado en el sistema.	
Poscondición:	El usuario accederá al sistema y tendrá privilegios específicos dependiendo del tipo de usuario, es decir, cada coordinador	

	solo tendrá acceso a su área asignada.
Presunción:	El sitio web y la base de datos están disponibles.

CASO DE USO 2

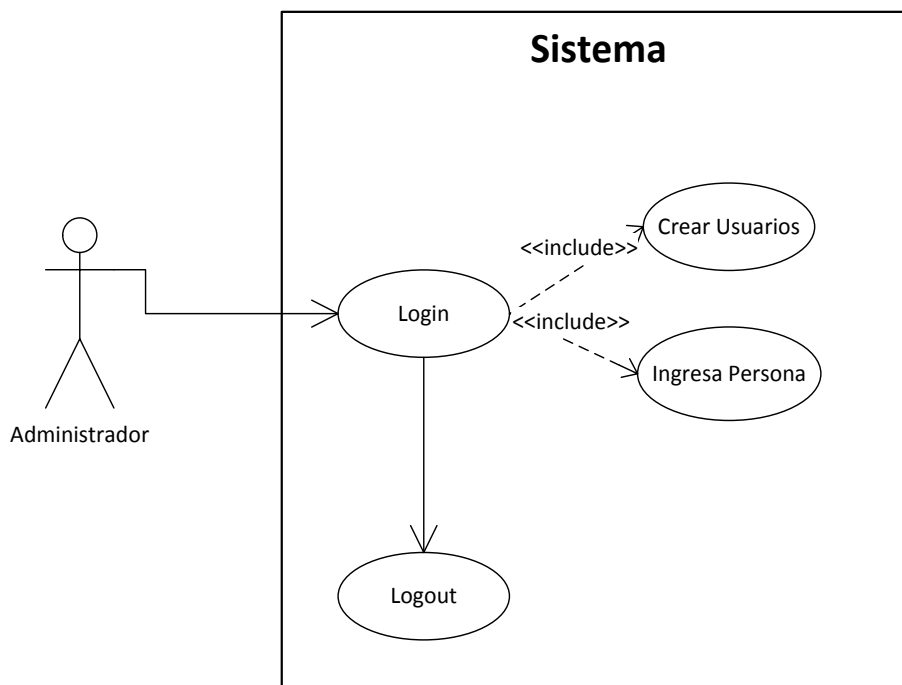


Figura 3.2. Caso de uso Administrador
Fuente: Las Autoras

Cuadro 3.5. Caso de uso 2/ CU_5, Administrador

Nombre:	Crear Usuarios/CU-5	
Actor(es):	Administrador	
Descripción:	Administra, controla, ingresa, habilita e inhabilita usuarios	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Estar autenticado en el sistema	
	2. Estar en la sección de crear usuarios	2. Muestra pantalla la sección de crear usuarios.
	3. Realiza un nuevo	3. Muestra pantalla de las opciones de ingreso.

	ingreso o habilita e inhabilita usuario	
	4. Finaliza o completa operación realizada	4. Guarda en la bd datos realizados.
Precondición:	El usuario debe estar autenticado en el sistema	
Poscondición:	Proveer el nombre de usuario y clave para el login respectivo	
Presunción:	El sitio web y la base de datos están disponibles.	

CASO DE USO 3

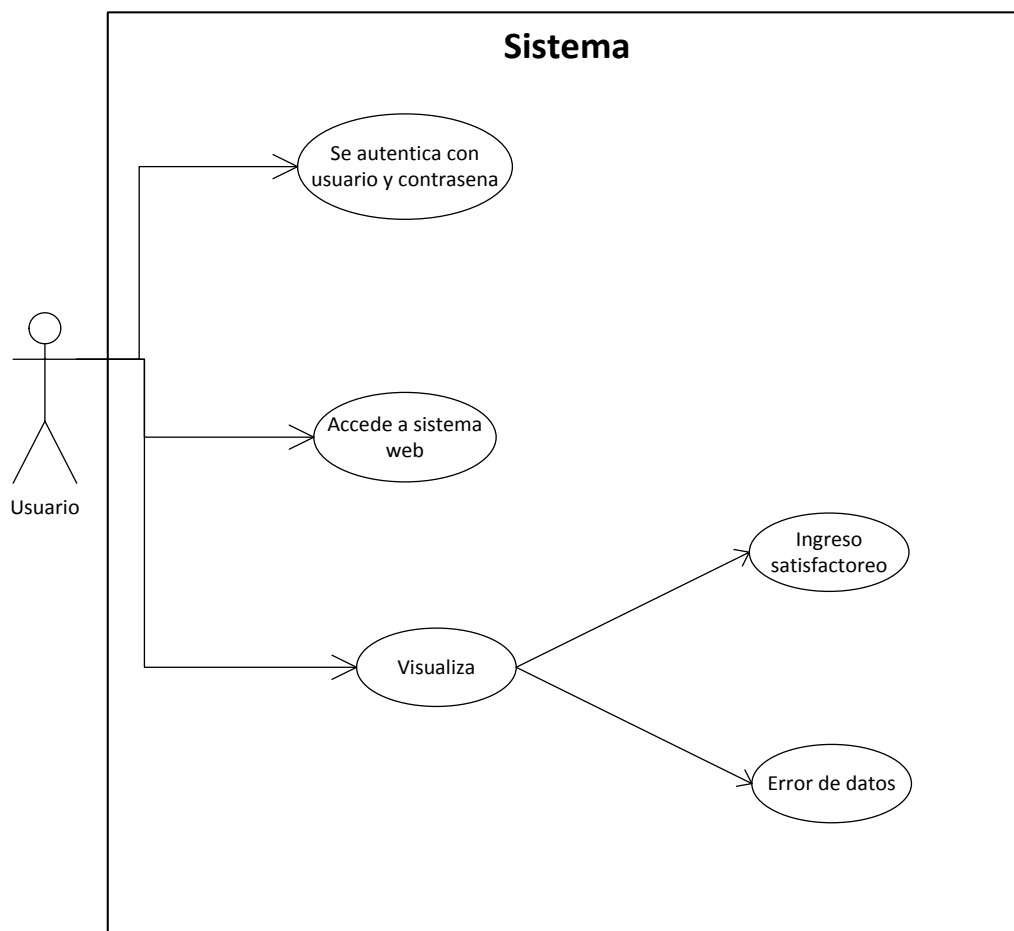


Figura 3.3. Caso de uso de Usuario
Fuente: Las Autoras

Cuadro 3.6. Caso de uso 3/ CU_6, Autenticación del usuario-Ingreso al sistema web

Nombre:	CU_6, Autenticación del usuario-Ingreso al sistema web	
Actor(es):	Director de Carrera Agrícola, Coordinador de áreas de producción	
Descripción:	La página de autenticación de usuarios debe permitir al Director de la Carrera Agrícola y Coordinadores de cada área de producción acceder al sistema web y realizar las funciones pertinentes a cada usuario.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Ingresar Usuario y Contraseña	Compara los datos ingresados con la base de datos.
	Accede al sistema correctamente o vuelve a intentarlo	Muestra un mensaje de Ingreso satisfactorio o Error de datos
	Visualiza la página con el mapa de uso	Muestra el Index
Precondición:	El usuario debe contar con un nombre de usuario y una contraseña asignados previamente para poder acceder a las opciones pertinentes en el sistema web.	
Poscondición:	El código del usuario se mantiene mientras su sesión esté abierta después de que se haya autenticado en el sistema.	
Presunción:	El sitio web y la base de datos están disponibles.	

CASO DE USO 4

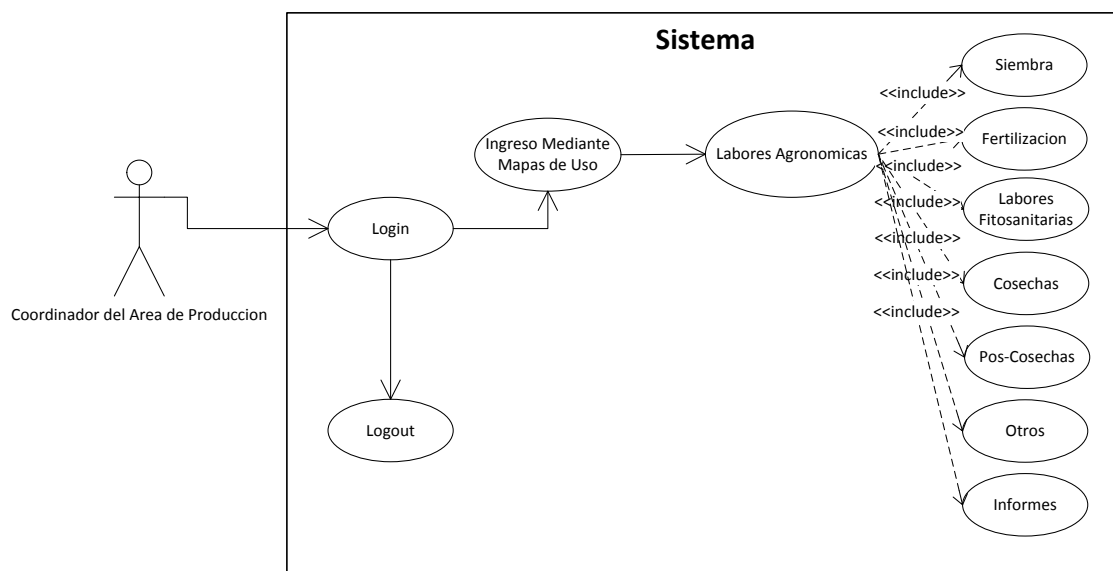


Figura 3.4. Caso de uso de Director de Carrera
Fuente: Las Autoras

Cuadro 3.7. Caso de uso 4/ CU_7, Generar Informes

Nombre:	CU_6, Generar Informes	
Actor(es):	Director de Carrera Agrícola	
Descripción:	Genera informes para cada una de las áreas de producción, para cada actividad, es decir, informe de siembra, de fertilización, y otros, e informe totales.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	Selecciona el área de producción	Muestra la pantalla con las opciones, por ingreso, mensual, semestral y anual.
	Selecciona Informe totales	Muestra un informa de todas las áreas de producción ya sea por ingreso, mensual, semestral y anual.
Precondición:	El usuario debe contar con un nombre de usuario y una contraseña asignados previamente para poder acceder a	

	las opciones pertinentes en el sistema web, este usuario debe ser de únicamente el que este asignado como Director de Agrícola.
Poscondición:	El código del usuario se mantiene mientras su sesión esté abierta después de que se haya autenticado en el sistema.
Presunción:	El sitio web y la base de datos están disponibles.

Una vez determinados los requerimientos se procedió a dar cumplimiento al diseño de un prototipo de la aplicación web para proporcionar al cliente una primera versión del producto.

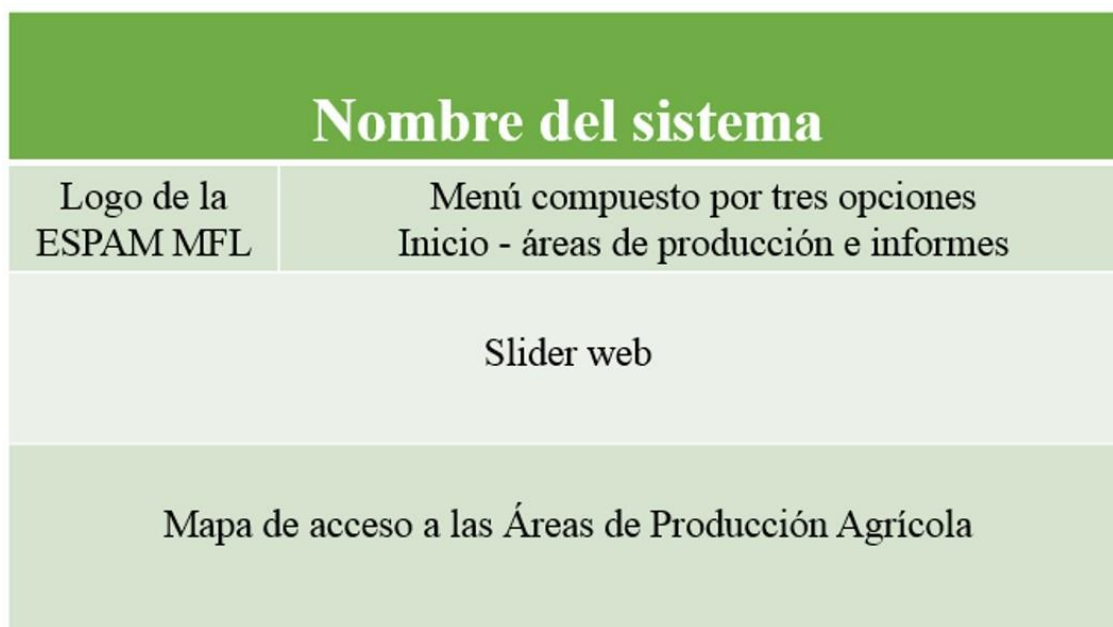


Figura 3.5. Prototipo de Ventana Inicio

Nombre del Sistema	
Nombre de Zona o Área	
Registro de Cultivo	Selección de Labor Agronómica

Figura 3.6 Prototipo de ventana de selección de labores agronómica

Nombre del Sistema
Ingreso de información de productos o insumos
Ingreso de datos para el control de bodega

Figura 3.7 Prototipo de ventana de ingreso de bodega

Concluida la segunda fase se desarrolla la tercera parte de la metodología.

3.1.1.3. MIDAS/DB

Es la tercera iteración, se desarrolla por un lado la base de datos web y por otro lado, se implementará una nueva versión del hipertexto y de la presentación con páginas dinámicas que extraen la información de la base de datos web (BD), dado que la navegación entre páginas normalmente implica una consulta a la BD.

Luego de la recopilación de los requerimientos se procedió al diseño de la base de datos que se elaboró con el gestor de base de datos SQL-SERVER 2008 R2 y cuyo diccionario de datos se detalla a continuación:

Cuadro 3.8. Diccionario de datos de la tabla “Archivo”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_archivo	Int	4	N	Y	N
Nombre	Nvarchar(500)	500	Y	N	N
Descripcion	Nvarchar(500)	500	Y	N	N
Fecha	Datetime	8	Y	N	N
Archivo	Varbinary(MAX)	-1	Y	N	N

La tabla archivo sirve para registrar documentos pdf de información meteorológica, sean publicaciones de INAMHI u otros, consta de 5 campos entre ellos id_archivo que es un código único para cada documento registrado y esta establecida como clave principal, de la misma manera se encuentran el campo Nombre donde se podrá registrar un título referente al archivo guardado, en el campo descripción se definirá en 500 caracteres a que se refiere dicho documento, así mismo se registrará la fecha de ingreso y finalmente el campo archivo servirá para alojar el documento pdf.

Cuadro 3.9. Diccionario de datos de la tabla “Áreas Agrícolas”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_areasagricolas	Int	4	N	Y	N
Nombre_areas	Nvarchar(100)	100	Y	N	N

La tabla áreas agrícolas servirá para registrar en nombre de cada área agrícola o zona y su respectivo id o clave única.

Cuadro 3.10. Diccionario de datos de la tabla “Bodega”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_bodega	Int	4	N	Y	N
NombrePE	Nvarchar(100)	100	Y	N	N
CantTotal	numeric(10, 2)	9	Y	N	N
id_tipo	Int	4	Y	N	N

NombreBodega	nvarchar(100)	100	Y	N	N
--------------	---------------	-----	---	---	---

En la tabla tb_bodega se registrara su id o código único, en nombre del equipo o insumo en en campo NombrePE, en CantTotal en numero de existencias, el id del tipo, es decir, si es equipo o insumo, 1 o 2 si es insumo o equipo respectivamente, finalmente en el campo NombreBodega se ingresara a que bodega pertenece dicho equipo o insumo, ya sea Genera, Cacao, Convencional, Frutales, Organica y Platanera.

Cuadro 3.11. Diccionario de datos de la tabla “Campos”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_campos	Int	4	N	Y	N
Id_labores	Int	4	Y	N	Y
Fecha	Datetime	8	Y	N	N
Responsables	nvarchar(50)	50	Y	N	N
Valores_Totales	numeric(10, 2)	9	Y	N	N
id_areasagricolas	Int	4	N	N	Y
id_producto	Int	4	N	N	Y
Observacion	Nvarchar(500)	500	Y	N	N

La tabla tb_campos permitirá registrar datos de cada labor que se realice, en id_campos se registrará el código único o clave primaria, en id_labores se registrata automáticamente el id de la labor a la que le corresponde sea siembra, cosecha y otros, lo mismo sucede con el campo id_areasagricolas automáticamente registra el id de acuerdo al área o zona a la que pertenece, asi mismo se registrara la fecha, en el campo Responsables ira el nombre de la persona encargada de la práctica sea un estudiante, semestre o docente, el campo Valores_Totales registra el monto total de dicha practica, el campo id_producto registrará el id del producto que se selecciono y en el campo Observación se podrá añadir una breve descripción en 500 caracteres de la practica.

Cuadro 3.12. Diccionario de datos de la tabla “Datos Meteorológicos”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_datosmet	Int	4	N	Y	N
Fecha	Datetime	8	Y	N	N
Viento	nvarchar(500)	500	Y	N	N
Precipitacion	nvarchar(10)	10	Y	N	N
TemperaturaMinima	nvarchar(10)	10	Y	N	N
TemperaturaMaxima	nvarchar(10)	10	N	N	N
HumedadRelativa	nvarchar(10)	10	N	N	N
Heliofania	nvarchar(10)	10	N	N	N
Evaporacion	nvarchar(15)	15	N	N	N
Nubosidad	nvarchar(15)	15	N	N	N

En la tabla Datos Meteorológicos se registrará diariamente los datos referentes al clima, entre ellos esta id o clave única, la fecha, el viento, precipitación, así mismo la Temperatura máxima y mínima que estará entre -45° y 45° , de la misma manera se registra la humedad relativa, la heliofania, evaporación y nubosidad, cada uno de estos campo registrará automáticamente el tipo de medida de acuerdo al campo.

Cuadro 3.13. Diccionario de datos de la tabla “Imagen”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_imagen	Int	4	N	Y	N
Nombre	Nvarchar(50)	50	Y	N	N
Imagen	Image	16	Y	N	N

La tabla Imagen permite almacenar en una imagen el calendario lunar anual, consta de tres campos, el id_imagen que es la clave primaria, el Nombre de la imagen y obviamente la imagen que contenga dicho calendario lunar.

Cuadro 3.14. Diccionario de datos de la tabla “Labores”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_labores	Int	4	N	Y	N
Nombre_Labores	Nvarchar(50)	50	Y	N	N

Esta tabla registrara el nombre de cada labor agronómica.

Cuadro 3.15. Diccionario de datos de la tabla “Persona”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_persona	Int	4	N	Y	N
CI	nvarchar(10)	10	Y	N	N
Nombres	nvarchar(50)	50	Y	N	N
Apellidos	nvarchar(50)	50	Y	N	N
Direccion	nvarchar(50)	50	Y	N	N
Telefono	nvarchar(50)	50	Y	N	N
Cargo	nvarchar(50)	50	Y	N	N

Esta tabla permite almacenar la información de los datos de una persona, su id, la Cedula de Ciudadania, Nombre, Apellido, Dirección, Telefono y el Cargo que tiene dentro de la institución.

Cuadro 3.16. Diccionario de datos de la tabla “Producto”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_producto	Int	4	N	Y	N
nom_producto	nvarchar(100)	100	Y	N	N
hectareas	numeric(10, 2)	9	Y	N	N
EpocaAnio	nvarchar(150)	150	Y	N	N
FechaRegistro	Datetime	8	Y	N	N

En la tabla producto se registrará el producto a tratar sea este cacao, maíz, frutas, entre otros, para ellos se registra el id_producto que es su clave primaria, el nombre del producto, el número de hectáreas en las que se encuentra dicho producto sembrado, la época del año y la fecha de registro.

Cuadro 3.17. Diccionario de datos de la tabla “Responsable

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_responsable	Int	4	N	Y	N
Nombresponsable	Nvarchar(50)	50	Y	N	N

En esta tabla se registrará el responsable de las lasbores agronómicas

Cuadro 3.18. Diccionario de datos de la tabla “Tipo”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_tipo	Int	4	N	Y	N
NombreTipo	Nvarchar(50)	50	Y	N	N

Esta tabla únicamente tendrá registrado si es equipo o insumo lo que se registra en la tabla bodega.

Cuadro 3.19. Diccionario de datos de la tabla “Usuario”

Columna	Tipo de Datos	Tamaño	Nulos	Clave Primaria	Clave Foránea
id_usuario	Int	4	N	Y	N
id_persona	Int	4	N	N	Y
Nom_usuario	nvarchar(50)	50	Y	N	N
Contraseña	nvarchar(50)	50	Y	N	N
Tipo_user	nvarchar(50)	50	Y	N	N

En la tabla usuario se registran los usuarios que podrán acceder al sistema, para ello se almacena un id o clave única, el id de la persona que se almacenara automáticamente al seleccionar el número de cédula de la persona, el nombre de usuario, la contraseña y el tipo de usuario, este será únicamente Administrador, Director, Cacao, Convencional, Frutales, Organica y Platanera.

Cuadro 3.20. Diccionario de datos de la tabla “transacciones”

Id_transaccion	int	4	N	Y	N
Precio	numeric(10, 2)	9	Y	N	N
UnidadMedida	nvarchar(150)	150	Y	N	N
Fecha	Datetime	8	Y	N	N
Financiado	nvarchar(50)	50	Y	N	N
CantNuevo	numeric(10, 2)	9	Y	N	N
Personal	nvarchar(50)	50	Y	N	N
Observacion	nvarchar(500)	500	Y	N	N
TipoTransaccion	nvarchar(50)	500	Y	N	N

La tabla transacción registrará el manejo de los equipos e insumos de bodega para ellos se registra el id_transaccion, el precio de dicho material, así mismo, la unidad de medida, la fecha, por quien esta financiado, ya sea por un estudiante, docente o universidad, la cantidad a usar o ingresar, la persona encargada de realizar dicha transacción, una descripción de 500 caracteres y el tipo de transacción que puede ser ingreso o egreso.

A continuación se muestra el Diagrama E/R de la base de datos:

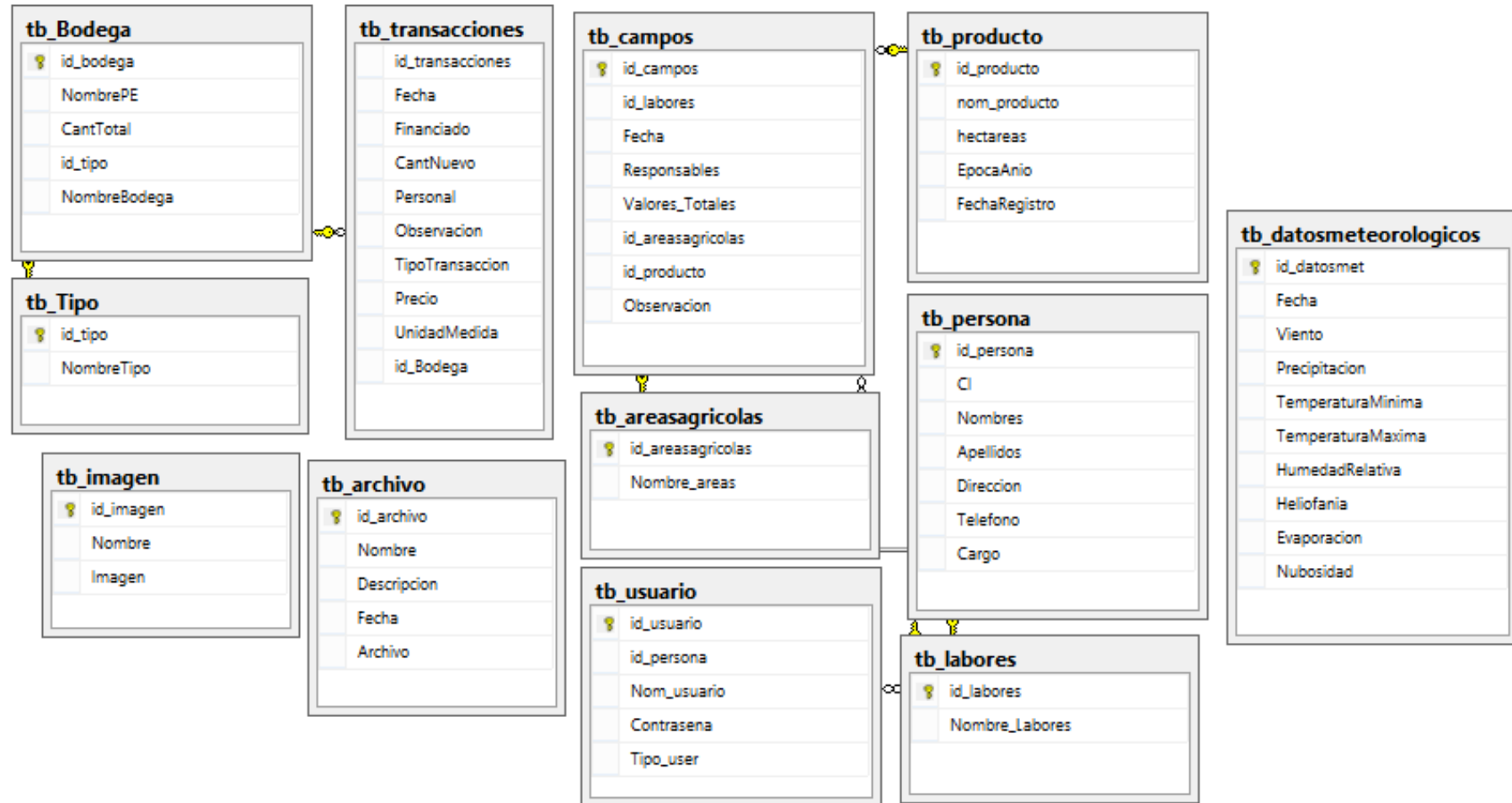


Figura 3.8. Diagrama E/R de la base de datos.

Una vez terminada la base de datos se llevó a cabo el diseño de la interfaz de usuario y la programación de cada uno de los formularios, se utilizó una programación en tres capas, la capa de acceso a datos, capa negocio y la capa presentación, a continuación se muestran las tres capas utilizadas en el sistema:

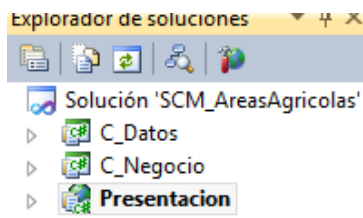


Figura 3.9. Captura de las tres capas usadas en el sistema.

Como se puede observar en la figura 3.8 en capa de acceso a datos la conexión se la realizo por DataSet:

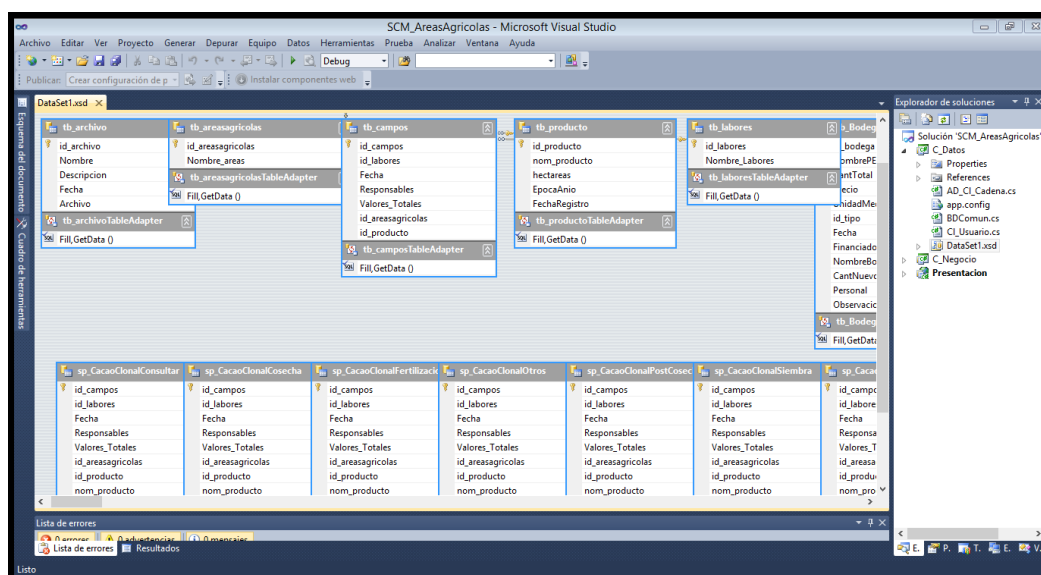


Figura 3.10. Modelo de base de datos en el DataSet de la Capa Datos

La capa negocio se desarrolló en el lenguaje C# utilizando la programación orientada a objetos, además se realizó una referencia a la capa datos para poder utilizar los métodos de la base de datos como se puede observar en el anexo 5.

A continuación se muestra el diagrama de clases de la capa negocio.

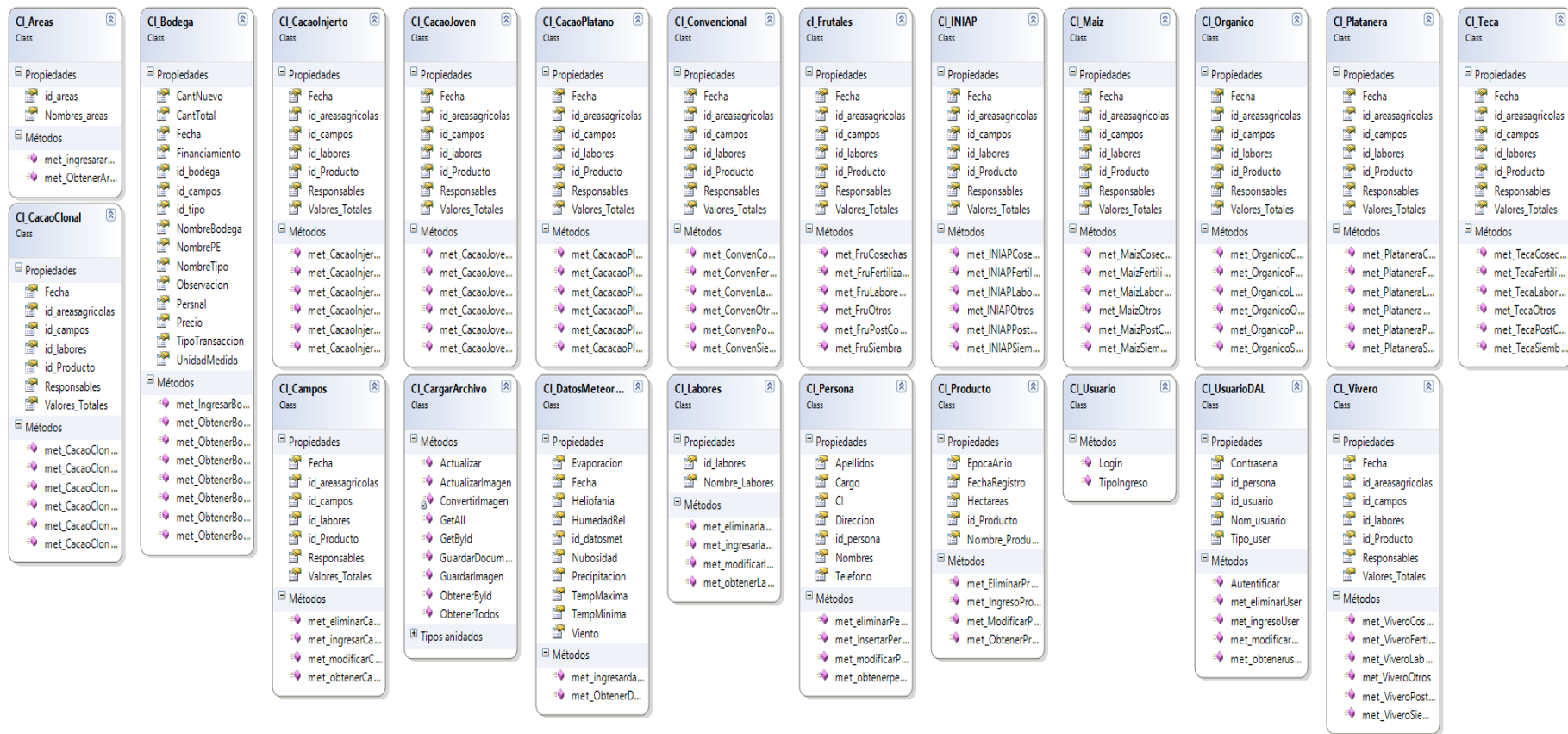


Figura 3.11. Diagrama de clases de la capa negocio

Otro software que se utilizó en el desarrollo del sistema es Adobe Flash Professional CS6, este es un programa que se usa normalmente para crear animaciones para páginas de internet, aunque las posibilidades son muchas: crear dibujos animados, presentaciones multimedia, aplicaciones móviles o de escritorio, páginas web completas, juegos, crear interactividad, menús y navegación para la web, etcétera. Marqués *et al.* (2013) menciona que sirve para crear efectos interactivos, mientras que Adobe (2014) señala que proporciona un entorno de edición para crear contenido de animación y multimedia para web, lo cual permite que los diseñadores visuales creen experiencias interactivas de aspecto similar en escritorios y dispositivos móviles.

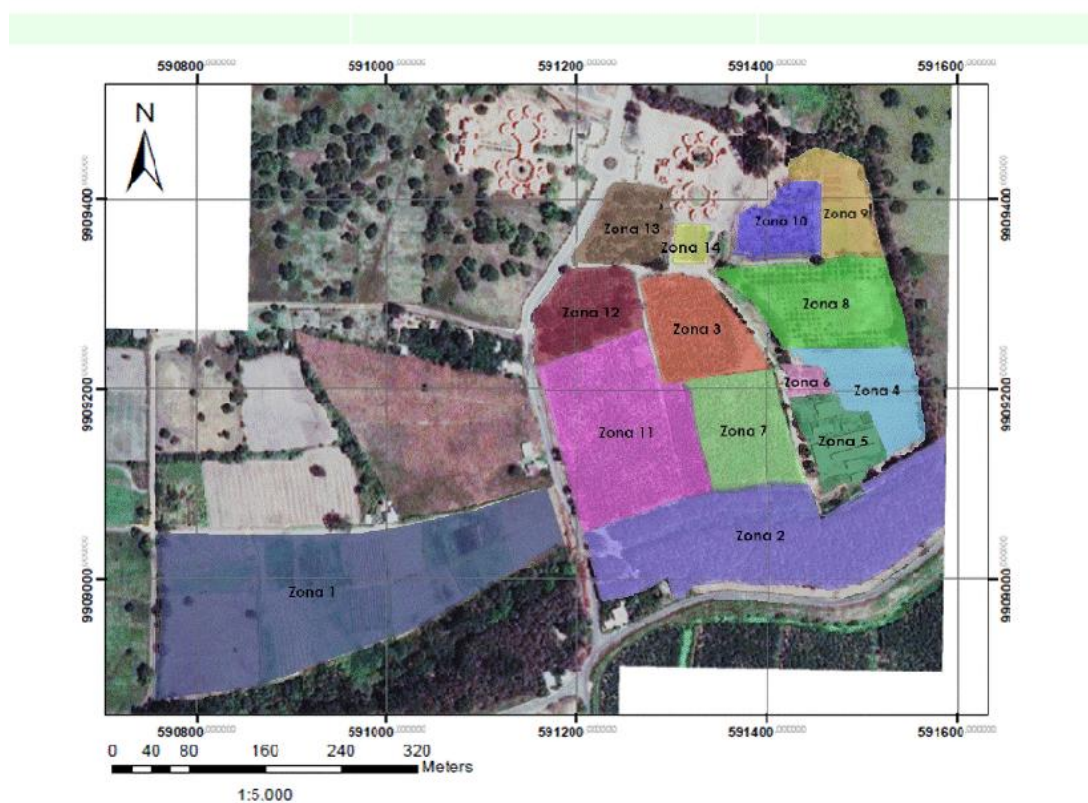


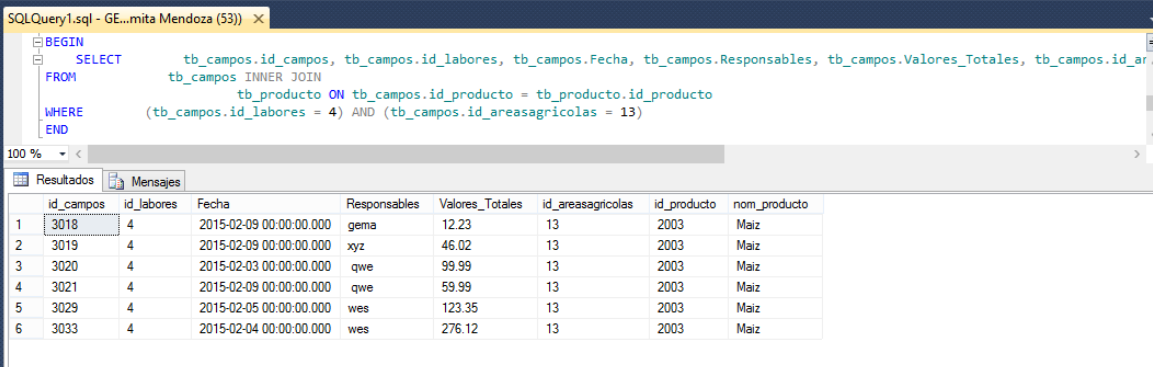
Figura 3.12. Vista del mapa realizado en Adobe Flash

3.1.1.4. MIDAS/FC

Esta es la fase final en la que se incorporarán los servicios dinámicos a la aplicación, por lo que es necesario realizar esta fase de acuerdo a los

requerimientos del usuario. En esta fase también es necesario realizar todas las pruebas necesarias antes de la implementación para que no se produzcan errores en el uso de la misma. Otro factor de fundamental importancia será brindar una pequeña capacitación a las personas que usaran el sistema para mejorar su desempeño en la aplicación.

Para dar paso al cumplimiento de esta fase de la metodología primeramente se procedió a realizar las pruebas necesarias en la base de datos para poder comprobar su funcionalidad, esto se lo corroboro mediante la ejecución de sentencias SQL en el sistema gestor de base de datos de Microsoft SQL server 2008 R2.



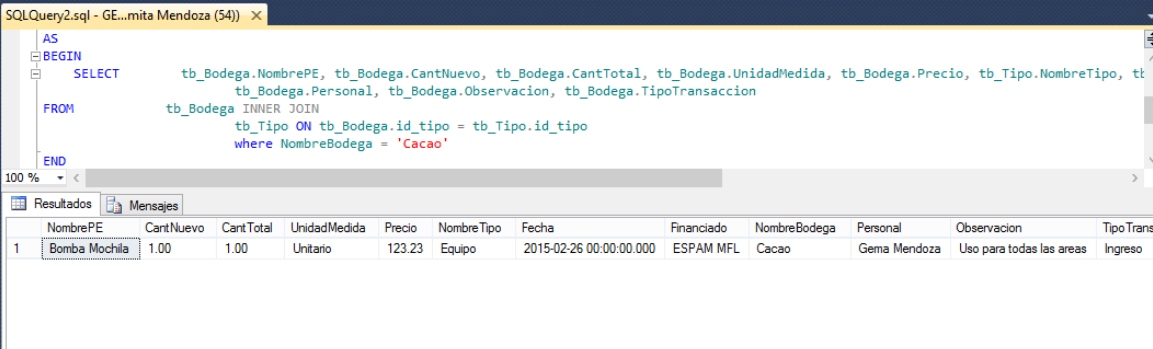
```

BEGIN
SELECT      tb_campos.id_campos, tb_campos.id_labores, tb_campos.Fecha, tb_campos.Responsables, tb_campos.Valores_Totales, tb_campos.id_ar
FROM        tb_campos INNER JOIN
           tb_producto ON tb_campos.id_producto = tb_producto.id_producto
WHERE       (tb_campos.id_labores = 4) AND (tb_campos.id_areasagricolas = 13)
END

```

id_campos	id_labores	Fecha	Responsables	Valores_Totales	id_areasagricolas	id_producto	nom_producto
1	3018	2015-02-09 00:00:00.000	gema	12.23	13	2003	Maiz
2	3019	2015-02-09 00:00:00.000	xyz	46.02	13	2003	Maiz
3	3020	2015-02-03 00:00:00.000	qwe	99.99	13	2003	Maiz
4	3021	2015-02-09 00:00:00.000	qwe	59.99	13	2003	Maiz
5	3029	2015-02-05 00:00:00.000	wes	123.35	13	2003	Maiz
6	3033	2015-02-04 00:00:00.000	wes	276.12	13	2003	Maiz

Figura 3.13. Consulta en SQL Server para obtener las cosechas del área cacao clonal.



```

AS
BEGIN
SELECT      tb_Bodega.NombrePE, tb_Bodega.CantNuevo, tb_Bodega.CantTotal, tb_Bodega.UnidadMedida, tb_Bodega.Precio, tb_Tipo.NombreTipo, t
           tb_Bodega.Personal, tb_Bodega.Observacion, tb_Bodega.TipoTransaccion
FROM        tb_Bodega INNER JOIN
           tb_Tipo ON tb_Bodega.id_tipo = tb_Tipo.id_tipo
WHERE       NombreBodega = 'Cacao'
END

```

NombrePE	CantNuevo	CantTotal	UnidadMedida	Precio	NombreTipo	Fecha	Financiado	NombreBodega	Personal	Observacion	TipoTransi	
1	Bomba Mochila	1.00	1.00	Unitario	123.23	Equipo	2015-02-26 00:00:00.000	ESPAM MFL	Cacao	Gema Mendoza	Uso para todas las areas	Ingreso

Figura 3.14. Consulta en SQL Server para obtener la bodega del área cacao.

```

GO
ALTER PROCEDURE [dbo].[sp_ConsultaPersona]
AS
BEGIN
SELECT      id_persona, CI, Nombres, Apellidos, Direccion, Telefono, Cargo
FROM        tb_persona

```

id_persona	CI	Nombres	Apellidos	Direccion	Telefono	Cargo
1	1312020447	Gema Teresa	Mendoza Vera	Canuto	0991375705	Estudiante

Figura 3.15. Consulta en SQL Server para obtener datos de Personas. Seguidamente se dio paso a la ejecución de la aplicación web para verificar su correcto funcionamiento.

Bienvenid@ : admin Área : Administrador [Cerrar Sesión](#)

Escuela Superior Politécnica Agropecuaria de Manabí
Manuel Félix López

Sistema de Control y Monitoreo de las Áreas de Producción Agrícola

Cedula: 1312020504
 Nombres: Maria Paula
 Apellidos: Zamora Mendoza
 Direccion: Calceta
 Telefono: 0989898989
 Cargo: Director

Guardar Datos ingresados correctamente

Id	CI	Nombres	Apellidos	Direccion	Telefono	Cargo
Modificar	1312020447	Gema Teresa	Mendoza Vera	Canuto	0991375705	Estudiante

Figura 3.16. Ingreso de Persona por el usuario Administrador.

Volver **BODEGAS**

Nombre	Cantidad Nueva	Nombre Bodega	Tipo Producto
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/> General <input type="radio"/> Cacao <input type="radio"/> Convencional <input type="radio"/> Frutales <input type="radio"/> Organica <input type="radio"/> Platanera <input type="radio"/> Equipo <input type="radio"/> Insumo

Guardar

Cantidad Total	Precio	Unidad de Medida	Fecha	Financiado	Personal	Observación	Tipo de Transacción
<input type="text"/>	<input type="text"/>	<input type="text"/>	dd/mm/aaaa	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/> Ingreso <input type="radio"/> Egreso

Guardar

Figura 3.17. Ingreso de Bodega por el usuario Administrador

3.2. TÉCNICAS

La técnica que se utilizó fue la entrevista, ya que con esta se recopiló la información necesaria para poder realizar un software que cumpla las necesidades de las áreas de producción agrícola, esta técnica se muestra en el anexo 1, la mencionada entrevista se la realizó al Director de Carrera Agrícola y, así mismo se empleó la observación, misma que ayudó a las autoras a socializarse con los procesos realizados en las áreas agrícolas, permitiendo entregar un buen sistema de control y monitoreo, para ello se hizo una entrevista al Coordinador de Campo como se muestra en el anexo 2.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

Las autoras luego de realizar la recolección, análisis y estructura de los requerimientos, se llevó a cabo la elaboración de 4 casos de usos para determinar las funcionalidades del sistema, de igual manera se diseñó la base de datos en SQL SERVER 2008 R2, teniendo un total de 12 tablas las cuales contienen información acerca de bodegas, datos meteorológicos, personas, productos, labores agronómicas y calendario agrícola, para el efecto fue indispensable trabajar en coordinación con el Director de la Carrera de Agrícola y el Técnico de Campo de las Áreas de Producción, personas que encargadas de la administración de dichas áreas.

La base de datos cuenta con un total de 13 tablas las cuales contienen información acerca de bodegas, datos meteorológicos, personas, productos, entre otros, diseño que se muestra en el desarrollo metodológico. La tabla tb_bodega tiene un total de 5 campos, mientras que las tablas tb_tipo, tb_campos, tb_producto, tb_persona, tb_areasagricolas, tb_usuarios, tb_labores, tb_transacciones, tb_responsables, tb_datosmeteorologicos, tb_archivo y tb_iamgen tienen un total de 2, 7, 5, 7, 2, 5, 2, 10, 2, 10, 5 y 3 campos respectivamente, cada una de estas tablas tiene un campo id como clave principal.

En el desarrollo de la base de datos se realizaron procedimientos almacenados, entre ellos está el de almacenar un archivo:

```
CREATE PROCEDURE [dbo].[sp_GuardarArchivo]
@Nombre nvarchar(500),
@Descripcion nvarchar(500),
@Fecha datetime,
@Archivo varbinary(MAX)
AS
BEGIN
    INSERT INTO tb_archivo
        (Nombre, Descripcion, Fecha, Archivo)
VALUES
    (@Nombre,@Descripcion,@Fecha,@Archivo)
END
```

De la misma manera se encuentra la consulta de labores agronómicas

```
SELECT
    tb_campos.id_campos, tb_campos.Fecha, tb_campos.Valores_Totales,
    tb_campos.Observacion, tb_producto.nom_producto AS Producto,
    tb_responsable.nombre_responsable, tb_labores.Nombre_Labores,
        tb_areasagricolas.Nombre_areas
    FROM
        tb_campos INNER JOIN
        tb_producto ON tb_campos.id_producto =
    tb_producto.id_producto INNER JOIN
        tb_labores ON tb_labores.id_labores =
    tb_labores.id_labores INNER JOIN
        tb_responsable ON tb_campos.id_responsable =
    tb_responsable.id_responsable INNER JOIN
        tb_areasagricolas ON tb_campos.id_areasagricolas =
    tb_areasagricolas.id_areasagricolas

    WHERE
        (tb_campos.id_areasagricolas = 12) and (tb_labores.id_labores
    = tb_campos.id_labores) and (tb_labores.id_labores = @id_labores ) and
    tb_campos.Fecha between @Fecha and @Fecha1";
```

También se puede encontrar el procedimiento almacenado para ingresar datos de bodega:

```
CREATE PROCEDURE [dbo].[sp_InsertarBodega]
@NombrePE nvarchar(100),
@CantTotal numeric(10, 2),
@id_tipo int,
@NombreBodega nvarchar(100)

AS
BEGIN
    INSERT INTO tb_Bodega
        (NombrePE, CantTotal, id_tipo, NombreBodega)
VALUES
    (@NombrePE,@CantTotal,@id_tipo,@NombreBodega)
END
```

Finalizado el diseño de la base de datos se procedió a realizar las interfaces del sistema bajo el lenguaje de programación C# y Visual Studio como Entorno de Desarrollo Integrado o IDE. Se trabajó bajo la programación orientada a objetos y se dividió la solución del sistema en tres capas. Destacando que el mapa de acceso a las áreas fue diseñado en Adobe Flash Professional CS6, con el objetivo de proporcionar al usuario un entorno animado creando una experiencia interactiva.

Terminado el diseño de las interfaces gráficas, se hizo uso de la entrevista como técnica, misma que fue dirigida al señor Director de la Carrera de Agrícola Ing. Lenin Vera Montenegro para corroborar que la interfaz cumpliera con las características necesarias, mostrándose agradable, fácil y de intuitiva interacción ante el usuario.

A continuación se presentan las interfaces más relevantes del sistema web, las mismas que fueron previamente presentadas y aprobadas por el Sr. Director de la Carrera de Agrícola.



The image shows a login form titled "Login" in a green header. On the left, there is a 3D illustration of a green stick figure holding a key to unlock a large padlock. To the right of the illustration are two input fields: "Usuario:" and "Contraseña:". Below these fields is a green button labeled "Iniciar Sesión".

Figura 4.2. Inicio de sesión del sistema

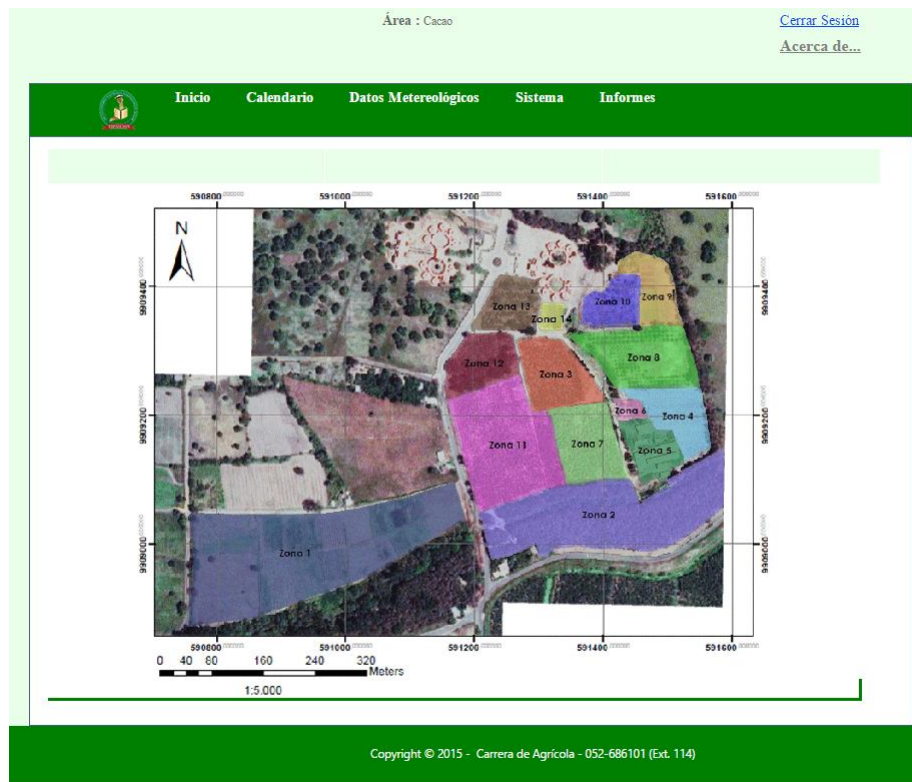


Figura 4.3. Mapa de acceso a las áreas

Al dar clic en inicio se muestra la ventana principal del sistema, compuesto por seis pestañas inicio, datos meteorológicos, calendario, áreas de producción, bodega e informes.

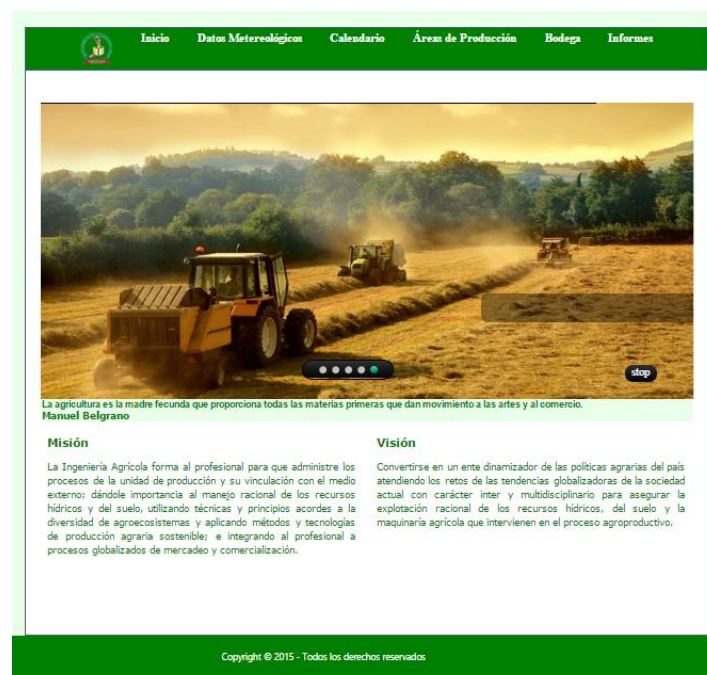


Figura 4.4. Ventana principal del sistema con su respectivo menú



Figura 4.5. Selección de labor agronómica



Figura 4.6. Registro de labor agronómica



The screenshot shows a web form titled "Registro Responsable" in green text. At the top center, there are two stylized human icons, a woman and a man, both wearing green shirts. Below the icons, the label "Responsable:" is followed by a white text input field. A green button with the white text "Guardar" is positioned directly below the input field.

Figura 4.7. Registro de responsable



The screenshot shows a web form titled "Registro de Cultivos Area Cacao Clonal" in green text. In the center, there is an illustration of a cacao tree with red pods and a broken pod showing white pulp and a brown seed. To the left of the illustration, there are four labels with corresponding input fields: "Fecha de Registro:" with a date format "dd/mm/aaaa", "Nombre Cultivo:", "Hectareas a Sembar:", and "Epoca del Año:". A green button with the white text "Guardar" is located at the bottom of the form.

Figura 4.8. Registro de cultivos

Así mismo se muestra el diseño de los reportes a emitir por parte de los encargados de las áreas de producción, brindando la finalidad de escoger el periodo y la labor de la cual se desea mostrar.

Área : Cacao [Cerrar Sesión](#)

[Inicio](#) [Calendario](#) [Datos Meteorológicos](#) [Sistema](#) [Informes](#)


Informe de Cultivos Area Cacao Clonal

Seleccionar las Fechas:

Desde:

Hasta:

Cargar



Escuela Superior Politécnica Agropecuaria de Manabí
Manuel Félix López
 Informe de Cultivos: Cacao Clonal
 Responsable: Ing. Paul Cedeño

Desde: 01/07/2015 Hasta: 31/07/2015

Fecha	nom_producto	Hectareas	Epoca/Año
01/07/2015 0:00:00	manzo	0.50	verano
19/07/2015 0:00:00	Maruja	1.00	verano
19/07/2015 0:00:00	camote	1.00	invierno
19/07/2015 0:00:00	mandarina	1.00	verano

Copyright © 2015 - Carrera de Agrícola - 052-686101 (Ext. 114)

Figura 4.9. Informes de áreas

Área : Cacao [Cerrar Sesión](#)
[Acerca de...](#)

[Inicio](#) [Calendario](#) [Datos Meteorológicos](#) [Sistema](#) [Informes](#)

Informes Cacao Clonal

Seleccionar la Labor Agrícola:


Siembra

Seleccionar las Fechas:

Desde:

Hasta:

Cargar



Escuela Superior Politécnica Agropecuaria de Manabí
Manuel Félix López
 Informe: Área Cacao Clonal
 Labor Agronómica: Siembra
 Responsable: Ing. Paul Cedeño

Desde: 01/07/2015 Hasta: 31/07/2015

Fecha	nombra_responsable	Valores_Totales	Producto	Observacion
01/07/2015 0:00:00	Jeniffer Parraga	47,00	Maruja	litre
01/07/2015 0:00:00	Jeniffer Parraga	47,00	camote	litre

Copyright © 2015 - Carrera de Agrícola - 052-686101 (Ext. 114)

Figura 4.10. Informes de Cultivos

Entre los códigos que se usaron para el desarrollo del sistema está el de consulta para generar reportes de cada una de las áreas de producción agrícola:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using C_Datos.DataSet1TableAdapters;
using System.Data.SqlClient;
using C_Datos;

namespace C_Negocio
{
    public class Cl_CacaoJoven
    {
        public int id_campos { get; set; }
        public int id_labores { get; set; }
        public DateTime Fecha { get; set; }
        public string nombre_responsable { get; set; }
        public string nom_producto { get; set; }
        public decimal Valores_Totales { get; set; }
        public int id_areasagricolas { get; set; }
        public int id_Producto { get; set; }
        public string Observacion { get; set; }
        public string Nombre_Labores { get; set; }
        public string Nombre_Areas { get; set; }
        public string Responsables { get; set; }

        public static List<CacaoJoven> met_CacaoClonalJoven(int id_lab, DateTime
    fech, DateTime fech1)
        {
            List<CacaoJoven> listacj = new List<CacaoJoven>();
            SqlConnection conn = AD_Cl_Cadena.ObtnerConexion();
            {

                string query = @"SELECT          tb_campos.id_campos,
    tb_campos.Fecha, tb_campos.Valores_Totales, tb_campos.Observacion,
    tb_producto.nom_producto AS Producto, tb_responsable.nombre_responsable,
    tb_labores.Nombre_Labores,
                tb_areasagricolas.Nombre_areas
                FROM          tb_campos INNER JOIN
                tb_producto ON tb_campos.id_producto =
    tb_producto.id_producto INNER JOIN
                tb_labores ON tb_labores.id_labores =
    tb_labores.id_labores INNER JOIN
                tb_responsable ON tb_campos.id_responsable =
    tb_responsable.id_responsable INNER JOIN
                tb_areasagricolas ON tb_campos.id_areasagricolas =
    tb_areasagricolas.id_areasagricolas

                WHERE          (tb_campos.id_areasagricolas = 12) and (tb_labores.id_labores
    = tb_campos.id_labores) and (tb_labores.id_labores = @id_labores ) and
    tb_campos.Fecha between @Fecha and @Fecha1";
```

```

        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@id_labores", id_lab);
        cmd.Parameters.AddWithValue("@Fecha", fech);
        cmd.Parameters.AddWithValue("@Fecha1", fech1);
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            CacaoJoven img = new
CacaoJoven(Convert.ToDateTime(reader["Fecha"]),
Convert.ToString(reader["Responsables"]),
Convert.ToDecimal(reader["Valores_Totales"]), Convert.ToString(reader["Producto"])
, Convert.ToString(reader["Observacion"]));
            listacj.Add(img);
        }
    }
    return listacj;
}

public class CacaoJoven
{
    public CacaoJoven( DateTime fecha, string responsable, decimal
Valores_Totales, string Producto, string Observacion)
    {
        this.Fecha = fecha;
        this.Responsables = responsable;
        this.Valores_Totales = Valores_Totales;
        this.Producto = Producto;
        this.Observacion = Observacion;
    }

    public CacaoJoven() { }
    public DateTime Fecha { get; set; }
    public string Responsables { get; set; }
    public decimal Valores_Totales { get; set; }
    public string Producto { get; set; }
    public string Observacion { get; set; }
}

public List<Cl_CacaoJoven> met_CacaoJovenSiembra()
{
    List<Cl_CacaoJoven> lista = new List<Cl_CacaoJoven>();
    sp_CacaoJovenSiembraTableAdapter dato = new
sp_CacaoJovenSiembraTableAdapter();
    foreach (var item in dato.GetData().ToList())
    {
        lista.Add(new Cl_CacaoJoven { id_campos = item.id_campos,
Nombre_Labores = item.Nombre_Labores, Nombre_Areas = item.Nombre_areas, Fecha =
item.Fecha, nom_producto = item.nom_producto, Valores_Totales =
item.Valores_Totales, Observacion = item.Observacion, nombre_responsable =
item.nombre_responsable });
    }
    return lista;
}

public List<Cl_CacaoJoven> met_CacaoJovenFertilizacion()

```

```

    {
        List<Cl_CacaoJoven> lista = new List<Cl_CacaoJoven>();
        sp_CacaoJovenFertilizacionTableAdapter dato = new
sp_CacaoJovenFertilizacionTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoJoven { id_campos = item.id_campos,
Nombre_Labores = item.Nombre_Labores, Nombre_Areas = item.Nombre_areas, Fecha =
item.Fecha, nom_producto = item.nom_producto, Valores_Totales =
item.Valores_Totales, Observacion = item.Observacion, nombre_responsable =
item.nombre_responsable });
        }
        return lista;
    }

    public List<Cl_CacaoJoven> met_CacaoJovenLaboresFitosanitarias()
    {
        List<Cl_CacaoJoven> lista = new List<Cl_CacaoJoven>();
        sp_CacaoJovenLaboresFitosanitariasTableAdapter dato = new
sp_CacaoJovenLaboresFitosanitariasTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoJoven { id_campos = item.id_campos,
Nombre_Labores = item.Nombre_Labores, Nombre_Areas = item.Nombre_areas, Fecha =
item.Fecha, nom_producto = item.nom_producto, Valores_Totales =
item.Valores_Totales, Observacion = item.Observacion, nombre_responsable =
item.nombre_responsable });
        }
        return lista;
    }

    public List<Cl_CacaoJoven> met_CacaoJovenCosecha()
    {
        List<Cl_CacaoJoven> lista = new List<Cl_CacaoJoven>();
        sp_CacaoJovenCosechaTableAdapter dato = new
sp_CacaoJovenCosechaTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoJoven { id_campos = item.id_campos,
Nombre_Labores = item.Nombre_Labores, Nombre_Areas = item.Nombre_areas, Fecha =
item.Fecha, nom_producto = item.nom_producto, Valores_Totales =
item.Valores_Totales, Observacion = item.Observacion, nombre_responsable =
item.nombre_responsable });
        }
        return lista;
    }

    public List<Cl_CacaoJoven> met_CacaoJovenPostCosecha()
    {
        List<Cl_CacaoJoven> lista = new List<Cl_CacaoJoven>();
        sp_CacaoJovenPostCosechaTableAdapter dato = new
sp_CacaoJovenPostCosechaTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoJoven { id_campos = item.id_campos,
Nombre_Labores = item.Nombre_Labores, Nombre_Areas = item.Nombre_areas, Fecha =
item.Fecha, nom_producto = item.nom_producto, Valores_Totales =
item.Valores_Totales, Observacion = item.Observacion, nombre_responsable =
item.nombre_responsable });
        }
        return lista;
    }

```

```

    }

    public List<Cl_CacaoJoven> met_CacaoJovenPostOtros()
    {
        List<Cl_CacaoJoven> lista = new List<Cl_CacaoJoven>();
        sp_CacaoJovenOtrosTableAdapter dato = new
sp_CacaoJovenOtrosTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoJoven { id_campos = item.id_campos,
Nombre_Labores = item.Nombre_Labores, Nombre_Areas = item.Nombre_areas, Fecha =
item.Fecha, nom_producto = item.nom_producto, Valores_Totales =
item.Valores_Totales, Observacion = item.Observacion, nombre_responsable =
item.nombre_responsable });
        }
        return lista;
    }
}
}
}

```

Se puede observar el código de cargar archivos e imágenes:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using C_Datos.DataSet1TableAdapters;
using System.Data.SqlClient;
using C_Datos;

namespace C_Negocio
{
    public class Cl_CargarArchivo
    {
        public static void GuardarDocumento(string nombre, string descripcion,
DateTime fecha, byte[] archivo)
        {
            SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();

            SqlCommand cmd = new SqlCommand("sp_GuardarArchivo", conexion);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@nombre", nombre);
            cmd.Parameters.AddWithValue("@descripcion", descripcion);
            cmd.Parameters.AddWithValue("@fecha", fecha);
            SqlParameter archParam = cmd.Parameters.Add("@archivo",
System.Data.SqlDbType.VarBinary);
            archParam.Value = archivo;
            cmd.ExecuteNonQuery();
        }

        public static void GuardarImagen(string nombre, byte[] archivo)
        {
            SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();

```

```

        SqlCommand cmd = new SqlCommand("sp_GuardarImagen", conexion);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@nombre", nombre);
        SqlParameter archParam = cmd.Parameters.Add("@imagen",
System.Data.SqlDbType.Image);
        archParam.Value = archivo;
        cmd.ExecuteNonQuery();

    }

    public class Archivo
    {
        public Archivo(int id_archivo, string nombre, string descripcion,
DateTime fecha)
        {
            this.id_archivo = id_archivo;
            this.Nombre = nombre;
            this.Descripcion = descripcion;
            this.Fecha = fecha;
        }

        public Archivo() { }

        public int id_archivo { get; set; }
        public string Nombre { get; set; }
        public string Descripcion { get; set; }
        public DateTime Fecha { get; set; }
        public byte[] Archivos { get; set; }
    }

    public static Archivo GetById(int Id)
    {
        Archivo arch = null;
        SqlConnection conn = BDComun.ObtnerConexion();
        {
            string query = @"top 10 SELECT id_archivo, Nombre, Descripcion,
Fecha, Archivo
                                FROM          tb_archivo
                                WHERE id_archivo = @id_archivo
                                ORDER BY id_archivo DESC";

            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@id_archivo", Id);

            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                arch = new Archivo(Convert.ToInt32(reader["id_archivo"]),
Convert.ToString(reader["Nombre"]), Convert.ToString(reader["Descripcion"]),
Convert.ToDateTime(reader["Fecha"]));
                arch.Archivos = (byte[])reader["Archivo"];
            }
        }
        return arch;
    }

    public static List<Archivo> GetAll()
    {
        List<Archivo> lista = new List<Archivo>();
    }

```

```

SqlConnection conn = BDComun.ObtnerConexion();
{
    string query = @"SELECT id_archivo,Nombre, Descripcion, Fecha
                    FROM      tb_archivo";
    SqlCommand cmd = new SqlCommand(query, conn);
    SqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        Archivo img = new
Archivo(Convert.ToInt32(reader["id_archivo"]),
Convert.ToString(reader["nombre"]), Convert.ToString(reader["descripcion"]),
Convert.ToDateTime(reader["fecha"]));
        lista.Add(img);
    }
}
return lista;
}

public class Imagen
{
    public Imagen(int id_image, string nombre)
    {
        this.id_imagen = id_image;
        this.Nombre = nombre;
    }

    public Imagen() { }

    public int id_imagen { get; set; }
    public string Nombre { get; set; }
    public byte[] Images { get; set; }
}

public static Imagen ObtenerById(int id_imagen)
{
    Imagen imagenes = null;

    SqlConnection conn = BDComun.ObtnerConexion();
    {
        string query = @"SELECT id_imagen,Nombre,Imagen
                        FROM tb_imagen WHERE id_imagen = @id_imagen";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@id_imagen", id_imagen);
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            imagenes = ConvertirImagen(reader);
        }
        conn.Close();
    }
    return imagenes;
}

public static List<Imagen> ObtenerTodos()
{
    List<Imagen> lista = new List<Imagen>();
    SqlConnection conn = BDComun.ObtnerConexion();
    {
        string query = @"SELECT id_imagen, Nombre, Imagen

```

```

        FROM tb_imagen";

        SqlCommand cmd = new SqlCommand(query, conn);
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            lista.Add(ConvertirImagen(reader));
        }
        conn.Close();
    }

    return lista;
}

private static Imagen ConvertirImagen(IDataReader reader)
{
    Imagen imagens = new Imagen();
    imagens.id_imagen = Convert.ToInt32(reader["id_imagen"]);
    imagens.Nombre = Convert.ToString(reader["Nombre"]);
    if (reader["Imagen"] != DBNull.Value)
        imagens.Images = (byte[])reader["Imagen"];

    return imagens;
}

public static Imagen Actualizar(Imagen imagin)
{
    SqlConnection conn = BDComun.ObtnerConexion();
    {
        string query = @"UPDATE tb_imagen SET
                        Nombre = @Nombre,
                        Imagen = ISNULL(@Imagen, Imagen)
                        WHERE id_imagen = ( select max(id_imagen) from
tb_imagen)";

        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Nombre", imagin.Nombre);
        SqlParameter imageParam = cmd.Parameters.Add("@Imagen",
System.Data.SqlDbType.Image);

        cmd.ExecuteNonQuery();
    }

    return imagin;
}

public static void ActualizarImagen(string nombre, byte[] imagen)
{
    SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();

    SqlCommand cmd = new SqlCommand("act", conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@nombre", nombre);
    SqlParameter archParam = cmd.Parameters.Add("@imagen",
System.Data.SqlDbType.Image);
    archParam.Value = imagen;
    cmd.ExecuteNonQuery();

}

}
}

```

Los formularios de cada área de producción también cuentan con un código de programación:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using C_Negocio;
using System.Text;

namespace Presentacion.F_Areas
{
    public partial class areaareaorganica : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

            Literal1.Text = CargarDatos();
        }
        public string CargarDatos()
        {
            Cl_Producto dato = new Cl_Producto();
            StringBuilder tabla = new StringBuilder();

            tabla.Append("<table artista='tb_campos' class='tb_campos'
rules='all'><thead><tr>");
            tabla.Append("<th>Nombre</th>");
            tabla.Append("<th>Hectareas</th>");
            tabla.Append("<th>Epoca de Año</th>");
            tabla.Append("<th>Fecha de Registro</th>");
            tabla.Append("</thead></tr>");
            tabla.Append("<tbody>");

            foreach (var item in dato.met_ObtenerProducto().ToList())
            {
                tabla.Append("<tr>");

                tabla.Append("<td>" + item.Nombre_Producto + "</td>");

                tabla.Append("<td>" + item.Hectareas + "</td>");
                tabla.Append("<td>" + item.EpocaAnio + "</td>");
                tabla.Append("<td>" + item.FechaRegistro + "</td>");
                tabla.Append("<tr>");
            }

            tabla.Append("</tbody>");
            tabla.Append("</table>");

            return tabla.ToString();
        }
    }
}

```



```

protected void Button1_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/siembraao.aspx");
}

protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/fertilizacionao.aspx");
}

protected void Button3_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/laboresfitosanitariasao.aspx");
}

protected void Button4_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/cosechasao.aspx");
}

protected void Button5_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/postcosechasao.aspx");
}

protected void Button6_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/otrosao.aspx");
}

protected void Button7_Click(object sender, EventArgs e)
{
    Response.Redirect("../areaorganica9/informesao.aspx");
}

protected void Button8_Click(object sender, EventArgs e)
{
    almacenardatos();
    limpiar();
}

void limpiar()
{
    TextBox1.Text = " ";
    TextBox2.Text = " ";
    TextBox3.Text = " ";
    TextBox4.Text = " ";
}

void almacenardatos()
{
    Cl_Producto ingreso = new Cl_Producto();
    Label15.Text = ingreso.met_IngresoProducto(new Cl_Producto {
Nombre_Producto = TextBox2.Text, Hectareas = Convert.ToDecimal(TextBox3.Text),
FechaRegistro = Convert.ToDateTime(TextBox1.Text), EpocaAnio = TextBox4.Text });
}

protected void Button9_Click(object sender, EventArgs e)
{
    Response.Redirect("../BodegaOrgnica.aspx");
}

```

```

    }

    protected void Button11_Click(object sender, EventArgs e)
    {
        string _open =
"window.open('../areaorganica9/informesareaorganica.aspx', '_newtab');";
        ScriptManager.RegisterStartupScript(this, this.GetType(),
Guid.NewGuid().ToString(), _open, true);
    }
}
}

```

El sistema tiene una opción de ingreso de calendario lunar, así mismo se puede observar dicha imagen ingresada, para esto se utilizó el siguiente código:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using C_Negocio;

namespace Presentacion.F_Meteorologia
{
    public partial class Cargarimagenydocumentos : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            if (FileUpload1.HasFile)
            {
                using (BinaryReader reader = new
BinaryReader(FileUpload1.PostedFile.InputStream))
                {
                    string ext =
System.IO.Path.GetExtension(this.FileUpload1.PostedFile.FileName);
                    if (ext == ".jpg" || ext == ".jpeg" || ext == ".gif" || ext
== ".png" || ext == ".JPG" || ext == ".JPEG" || ext == ".GIF" || ext == ".PNG")
                    {
                        byte[] image =
reader.ReadBytes(FileUpload1.PostedFile.ContentLength);
                        Cl_CargarArchivo.GuardarImagen(FileUpload1.FileName, image);
                        Button4.Visible = true;
                        Button2.Visible = false;
                        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Imagen Cargada Correctamente" +
"');</script>");
                    }
                    limpiar();
                }
            }
        }
    }
}

```

```

        else
        {
            this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + " Formato de archivo incorrecto. Solo se
permiten imágenes con formato .jpg -- .jpeg -- .gif -- .png" + "');</script>");
        }
    }
    else
    {
        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Todos lo campos son obligatorios" +
"');</script>");
    }
}

protected void Button3_Click(object sender, EventArgs e)
{
    GuardarPdf();
}

private void GuardarPdf()
{
    if (FileUpload2.HasFile && TextBox2.Text != "")
    {
        using (BinaryReader reader = new
BinaryReader(FileUpload2.PostedFile.InputStream))
        {
            string ext =
System.IO.Path.GetExtension(this.FileUpload2.PostedFile.FileName);
            if (ext == ".pdf" )
            {
                byte[] image =
reader.ReadBytes(FileUpload1.PostedFile.ContentLength);
                Cl_CargarArchivo.GuardarDocumento(FileUpload2.FileName,
TextBox2.Text, Convert.ToDateTime(System.DateTime.Now), image);
                this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Archivo cargado correctamente" +
"');</script>");
                limpiar();
            }
            else
            {
                this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + " Formato de archivo incorrecto. Por
favor ingrese un archivo con formato .pdf" + "');</script>");
            }
        }
    }
    else
    {
        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Todos lo campos son obligatorios" +
"');</script>");
        limpiar();
    }
}

private void limpiar()
{
    TextBox2.Text = string.Empty;
}

```

```

    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        ActualizarImagen();
    }
    private void ActualizarImagen()
    {
        if (FileUpload1.HasFile)
        {
            using (BinaryReader reader = new
BinaryReader(FileUpload1.PostedFile.InputStream))
            {
                string ext =
System.IO.Path.GetExtension(this.FileUpload1.PostedFile.FileName);
                if (ext == ".jpg" || ext == ".jpeg" || ext == ".gif" || ext
== ".png" || ext == ".JPG" || ext == ".JPEG" || ext == ".GIF" || ext == ".PNG")
                {
                    byte[] image =
reader.ReadBytes(FileUpload1.PostedFile.ContentLength);
                    Cl_CargarArchivo.ActualizarImagen(FileUpload1.FileName,
image);

                    this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Imagen actualizada correctamente" +
"');</script>");

                    limpiar();
                }
                else
                {
                    this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + " Formato de archivo incorrecto. Solo se
permiten imágenes con formato .jpg -- .jpeg -- .gif -- .png" + "');</script>");
                }
            }
        }
        else
        {
            this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Todos lo campos son obligatorios" +
"');</script>");
            limpiar();
        }
    }

    protected void Button5_Click(object sender, EventArgs e)
    {
        Response.Redirect("/IndexMeteorologia.aspx");
    }
}
}

```

Una vez concluido el desarrollo del sistema web, se procedió a la verificación del mismo para ello se alojó en los servidores de la ESPAM MFL comprobando su funcionalidad; finalmente el Director de la Carrera de Agrícola Ing. Leonardo Vera, corroboró mediante un certificado que se puede observar en el anexo 6 que el sistema cumplió con los requerimientos propuestos.

En el cuadro 4.1. Se muestra la comparación al realizar el ingreso y reportes de las áreas de producción de forma manual y con sistema. Se observa claramente que con la implementación del sistema de control de monitoreo, el ingreso de información y la emisión de reportes se realizó en menor tiempo lo que corrobora la eficiencia del sistema.

Actividad	Manual	Sistema Informático	Optimización	
			Min/Seg	Porcentaje
Ingresos	00:01:50	00:00:13	00:01:37	91,33%
Consulta	00:03:00	00:00:20	00:02:80	93,33%
Reportes	00:06:10	00:00:35	00:05:75	94,26%
			Total:	92,98%

Cuadro 4.1: Comparación de los procesos con el antes y después del sistema
Fuente: Las Autoras

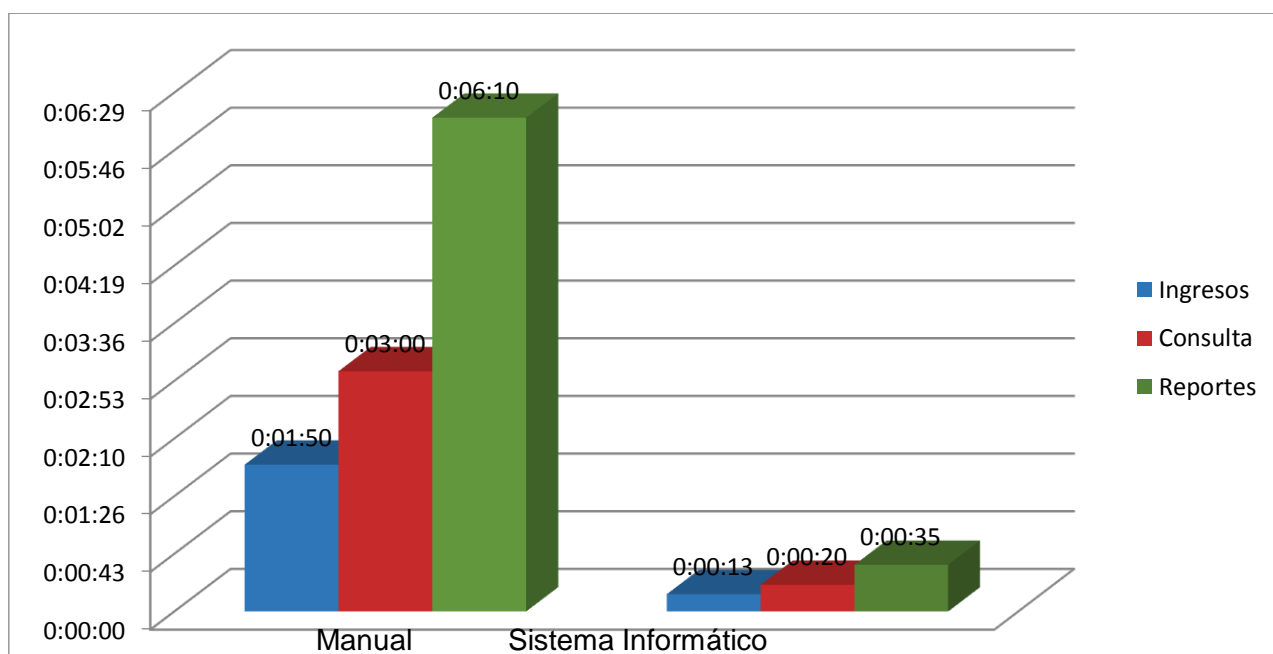


Figura 4.11. Cuadro comparativo del rendimiento del sistema

4.2. DISCUSIÓN

Las herramientas informáticas son realmente necesarias para el progreso de toda organización y mejoramiento de la calidad humana lo que corrobora Weisman (2003).

Existen software de control de cultivos a nivel mundial, gratis y listo para usar, tales como AgroWin®, que es un software agrícola 4.0 que permite llevar un registro y seguimiento detallado de las labores realizadas en cada lote o cultivo, permitiendo hacer seguimiento de labores realizadas, generar informes y estadísticas de labores. Beneficios similares ofrece AGROCON, este software está enfocado a los profesionales del campo que quieran controlar una explotación agraria al dedillo, sin tener que preocuparse de la cantidad de parcelas o tipo de cultivo controlando todo tipo de gastos e ingresos hasta un seguimiento pluviométrico de la zona, todo ello pasando por datos de parcelas, proveedores, códigos de gastos y productos, almacén, etc. Sin embargo por el hecho de ser sistemas informáticos genéricos suelen obviar ciertas necesidades específicas del usuario final, conllevando al usuario a verse en la necesidad de desarrollar aplicaciones que se ajusten a sus necesidades.

Precisamente esta situación se evidenció en la Carrera de Agrícola con el control y monitoreo de las áreas de producción de la ESPAM MFL, por lo que las autoras de la tesis consideraron pertinente la creación de un sistema que permita llevar un correcto control y monitoreo en cada una de ellas, facilitando un análisis concreto de lo que en realidad se usa para cada producción, y dando respuesta a las exigencias de los procesos productivos en cuanto a rendimiento, calidad, eficiencia y eficacia. Utilizando el sistema de AGROCON 3.0 y el sistema para la Carrera de Agrícola, se pudo evidenciar que las necesidades cubiertas son las mismas, tomando en cuenta que el sistema realizado por las autoras está basado en los requerimientos del cliente beneficiándolo de una mejor manera.

The screenshot displays the AGROCON 3.0 software interface for agricultural management. The main window is titled 'AGROCON - 3.0d (11-05-2010) Año 2010 - Gestión de Explotaciones Agrícolas'. The interface is divided into several sections:

- EXPLANTACIONES:** Lists agricultural operations with columns for 'Núm', 'Nombre o Denominación', 'Superf', and 'Plantas'. It shows two entries: '1 EXPLANTACIONES AGRICOLAS DEL SUR S.A.' and '2 LAS CAÑADAS REALES S.L.'.
- FINCAS o PARCELAS:** Lists land parcels with columns for 'Núm', 'Nombre o Denominación', 'Superf', and 'Plantas'. It shows two entries: '1 LOS LLANOS DEL MARQUES' and '2 LA HUERTA DE LA CAÑADA'.
- RECINTOS SIGPAC:** Lists SIGPAC plots with columns for 'Núm', 'Prov.', 'Munic.', 'Pol.', 'Parc.', 'Recint', 'Superficie', 'Plantas', and 'Localid.'. It shows two entries: '1 18 132 005 036 001 7.15 1112 PINOS' and '2 18 132 005 036 002 3.25 720 PINOS'.
- GASTOS/INGRESOS:** A table showing financial transactions. The table has columns: Fecha, Concepto, Precio, Cantidad, Importe, IVA, Total, and Grn. It lists various expenses like 'Talar', 'BASAGRAN L', 'SULFOCUPRIL', 'Recibo de agua', 'Seguros Sociales de Septiembre', 'Recibo electricidad 09/2009', 'I.B.I. año 2008', 'AMONIACO', 'Devolución cuotas 05/2008', 'ZOTAL', and 'Gasolina 98'.
- PRODUCCIÓN:** A table showing production records. The table has columns: Fecha, Producto, Variedad, Calidad, Calibre, Cantidad, Rendto, Cent u/Rdt, and Precio. It lists production of 'Aceitunas' in 'ARBOL' and 'SUELO' varieties.

The interface includes navigation buttons like 'Nuevo', 'Modificar', 'Borrar', and 'Imp', as well as summary statistics at the bottom of each section.

Figura 4.12. Vista del sistema AGROCON 3.0

El sistema implementado en las áreas de producción permitió agilizar los procesos en un 92,98% comparado con los procesos realizados anteriormente en dichas instalaciones, y la emisión de reportes solicitados por el Director de Carrera sea 100% real y confiable, cumpliendo con los objetivos fijados en la investigación. Sin embargo con la creación de nuevas áreas de producción pueden existir nuevos requerimientos para ello se debe realizar feedback, que es una técnica general para realizar retroalimentación y así poder mejorar cualquier sistema informático.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- ✓ El levantamiento de información permitió definir con exactitud los requerimientos funcionales. .
- ✓ El diseño de casos de uso y prototipos facilitó el desarrollo del sistema.
- ✓ En el desarrollo del sistema la aplicación de metodología en tres capas y programación orientada a objetos permitió organizar y estructurar el código eficientemente.
- ✓ La implementación del sistema permitió a los encargados de las áreas de producción manejar eficaz y eficientemente la información de las áreas de producción.

5.2. RECOMENDACIONES

- ✓ Realizar un análisis profundo a la información facilitada por las instituciones para definir correctamente los requerimientos finales y que el sistema cumpla con las expectativas deseadas.
- ✓ Diseñar el primer prototipo del sistema para determinar si se ajusta a los requerimientos y así avanzar confiadamente en el desarrollo del mismo.
- ✓ Elegir herramientas y programas informáticos que muestren un excelente nivel de seguridad, administración y facilidad de empleo, a fin de mantener la información íntegra, confiable y disponible.
- ✓ A los encargados del manejo del sistema realizar una capacitación previa a su uso, a fin de evitar errores en el manejo de la información.

BIBLIOGRAFÍA

- Abalde, A. y Muñoz, J. 2011. Metodología Cuantitativa Vs. Cualitativa. Formato PDF. En línea. Consultado el 5 de agosto de 2014. Disponible en <http://ruc.udc.es/dspace/bitstream/2183/8536/1/CC-02art7ocr.pdf>
- Adobe, 2014. ADOBE® FLASH® PROFESSIONAL, Ayuda y tutoriales. Formato PDF. En línea. Consultado el 17 de febrero de 2015. Disponible en https://helpx.adobe.com/es/pdf/flash_reference.pdf
- Agrocalidad (Agencia Ecuatoriana de Aseguramiento de la Calidad del Agro), 2008. En línea. Consultado el 4 de junio de 2014. Disponible en <http://www.agrocalidad.gob.ec/sanidad-vegetal/>
- Aibar, E. 2008. Las culturas de Internet: la configuración sociotécnica de la red de redes. Revista iberoamericana de ciencia tecnología y sociedad. Buenos Aires, AR. Vol 4. Nº 11.
- Álvarez, E y Montes, M. 2010. Sistema de gestión de requisitos para titulaciones en la Universidad de Cuenca.
- Arana, R. 2011. Monitoreo y evaluación: definición, funciones y usos. Formato PDF. En línea. Consultado el 02 de junio de 2014. Disponible en http://www.pasca.org/userfiles/M1_T7_ARANA_NI.pdf
- Bastarache, I; Fernández, B; Hernández, D; Del Valle, L; Duque, E. 2012. Sistema para la actualización de aplicaciones médicas. Revista Cubana de Informática Medica. Cuba.
- Buitrago, B. 2010. El lenguaje de programación comunicación programador y computadora. Journal Boliviano de Ciencias. Cochabamba. Vol 7. N 21. p 60 – 62.
- Cáceres, L. y Pinto, M. 2011. Modelo de programación asíncrona para Web transaccionales en un ambiente distribuido. Ingeniare. Revista chilena de ingeniería. Arica, Ch. Vol. 19. N 1. p 26 – 39.
- Carranza, E. y Ganchozo, E. 2013. Sistema web de gestión administrativa en el departamento de tesorería del cuerpo de bomberos “Casimiro Farfán”

del Cantón Bolívar. Tesis. Ing. Informática. ESPAM MFL. Calcuta-Manabí, EC. p 22.

De la Fuente, E. y Suárez, S. 2008. Problemas ambientales asociados a la actividad humana: la agricultura. Córdoba, Ar. Ecología austral. Vol. 18. N 3. p 239-252

Díaz, J. 2009. Las metodologías ágiles como garantía de calidad del software. REICIS. España. Vol 5. N 3. p 40-43.

Dominguez, C; Geijo, S; Sanchez, I; Imaz, C; Cabus, G. 2012. Revisión de los mecanismos implicados en el uso problemático de Internet. Revista de la Asociación Española de Neuropsiquiatría. Madrid, ES. Vol 32. Nº 116.

ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López), 2015. Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López. En línea. Consultado el 18 de mayo de 2015. Disponible en www.espam.edu.ec

Fernández, L. 2005. Un sondeo sobre la práctica actual de pruebas de software en España. REICIS. Madrid, Es. vol 1. N 002. p 43-54.

Franco, L. 2011. Concepto y Principio de Control. Formato PDF. En línea. Consultado el 20 de julio de 2014. Disponible en <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r93941.PDF>

Gomes, W. y De Faria, R. 2012. Calidad de los sitios Web en la percepción de los usuarios una Aplicación en los hoteles de João Pessoa, Paraíba, Brasil. Estudios y perspectivas en turismo. Buenos Aires, Ar. vol.21. N 4.

Jabba, D; Alcocer, A; Rojas, C. 2004. Análisis comparativo de las herramientas de programación Web: PHP, ASP Y JSP, bajo los sistemas operativos Linux y Windows. Ingeniería y desarrollo. 16 ed. Colombia. p 104-115.

López, B; Hernández, B; Sánchez, M; Cruz, E; Cano, R. 2009. PROGRAMACIÓN FÁCIL en el proceso enseñanza-aprendizaje. Innovación Educativa. México. Vol 9. N 48. p 61 - 71.

- Martinez, Y; Vega, L; Ferras, Y. 2014. Concepción del módulo de pacientes para el Sistema de Información Radiológica alas RIS. La Habana, CU. Revista Cubana de Informática Médica. Vol. 6. N° 2.
- Moreno, M; González, G; Echartea, D. 2008. Evaluación de la Calidad en Uso de Sitios Web Asistida por Software: SW – AQUA. Revista Avances en Sistemas e Informática. Colombia. Vol 5. N 1. p 147 – 154.
- Rodríguez, O; Cánovas, R; Infante, A; Ortíz, R; Pérez, A. 2013. SISCORFI: una aplicación web para el control de los recursos fitogenéticos. Cultivos Tropicales. La Habana, Cu. Vol 34. N 3. p 25- 31.
- Hidalgo, Y. y Rodriguez, R. 2013. La web semántica: una breve revisión. Revista Cubana de Ciencias Informáticas. La Habana-Cuba. Vol 7. N 1.
- Invest Manabi, s.f. Agronegocios. En línea. Consultado el 5 de junio de 2014. Disponible en http://investmanabi.com/index.php?option=com_content&view=article&id=24&Itemid=15
- LOES (Ley Orgánica de Educación Superior), 2010. Ley Orgánica de Educación Superior. RO N° 298. Formato PDF.
- Marques, N; Moreira, L; Moura, T; Nunes, L; Cândido, J; Áfio, J. 2013. Construction of an educational technology for teaching about nursing on peripheral venipuncture. Revista Gaúcha de Enfermagem. Porto Alegre, Br. Vol 34. N 2
- Moreano, E. 2014. Análisis, diseño, construcción e implementación de un portal de información y venta de aplicaciones de entrenamiento para el club de video juegos de la ESPE. Sandolquí, EC. Revista Tecnológica ESPE. p 4.
- Oliveros, A; Del Valle, S; Wehbe, R. Rousselot, J. 2011. Requerimientos para aplicaciones web. En línea. Consultado el 4 de Junio de 2014. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/20125>.
- Rivera, M. y Zambrano, F. 2012. Sistema on-line de pedidos en la farmacia 7 de agosto del Cantón Chone. Tesis. Ing. Informática. ESPAM MFL. Calceta-Manabí, EC. p 36.

- Rob, P. y Coronel, C. 2004. Sistemas de bases de datos: Diseño, implementación y administración. 5th Ed. Mexico. p 526-527.
- Rodríguez, P; Musat, D; Yagüe, A; Turhan, B; Rohunen, A; Kuvaja, P; Oivo, Markku. 2010. Adopción de metodologías ágiles: un estudio comparativo entre España y Europa. REICIS. España. Vol 6. N 4. p 6-28.
- Salazar, M; Vallejo, D; De la Torre, A; Campaña, M. 2012. Análisis, diseño e implementación de un sistema de control del catastro turístico para la empresa pública metropolitana de gestión del destino turístico de la ciudad de Quito. Revista ESPE.
- Santos, M; García, M; Cabrales, G; Castro, R; Santiesteban, C; Báez, E. 2006. Ronbinson más Citelivepro 2.0: programa para la organización bibliográfica. Biblios. Perú. Vol 7. N 24.
- SENPLADES (Secretaria Nacional de Planificación y Desarrollo), 2013. Plan Nacional para el Buen Vivir 2013-2017. Formato PDF.
- SENPLADES (Secretaria Nacional de Planificación y Desarrollo), 2012. Transformación de la Matriz Productiva. 1ed. Formato PDF.
- Stair, R. y Reynolds, W. 1999. Principios de sistemas de información: Enfoque administrativo. 4th ed. México. p 169-179.
- Stangarlin, L; Da Silva, F; Rosa, M. 2013. Desenvolvimento de uma ferramenta para construir dinamicamente Relatórios no crystal reports. Brasil. Revista Colloquium Exactarum. Vol. 5. p 26 - 29.
- Tello, J. 2013. La agricultura como sistema. Idesia. Arica, Ch. Vol. 31. N 1. p 3 – 4.
- Vidal, C; Hernández, D; Pereira, C; Del Rio, M. 2012. Aplicación de la Modelación Orientada a Aspectos. Información Tecnológica. Chile. Vol. 23. N 1. p 3 – 12.
- Weisman, K. 2003. Los pros y los contras del uso de la tecnología. En línea. Consultado el 28 de febrero de 2015. Disponible en

<http://portal.educ.ar/debates/educacionytic/nuevos-alfabetismos/los-pros-y-los-contras-del-uso-de-la-tecnologia-en-el-aprendizaje.php>.

Zapata, C. y Chaverra, J. 2010. Una mirada conceptual a la generación automática de código. EIA. Colombia. N 13. p 143-154.

Zapata, C. y Sierra, J. 2005. Medusa: una aplicación de la integración entre PDF y ASP.NET ®. DYNA. Colombia. vol 72. N 146. p 89-102.

Zapata, C. y Tamayo, P. 2009. Generación del diagrama de casos de uso a partir del lenguaje natural o controlado: Una revisión crítica. Vol. 76. p 193-203.

Zorrilla, M; Vela, B; Marcos, E. 2007. Una Aproximación Dirigida por Modelos para Diseñar y Construir Esquemas XML: Un Caso de Estudio. Revista Avances en Sistemas e Informática. Colombia. Vol 4. N 3. p 5-14.

ANEXOS

ANEXO 1

ENTREVISTA REALIZADA EL DIRECTOR DE LA CARRERA DE AGRÍCOLA

NOMBRE DE ENTREVISTADO: Ingeniero Lenin Vera

CARGO QUE OCUPA EN LA INSTITUCIÓN: Director de la Carrera de Agrícola.

FECHA Y HORA: 31 de julio de 2014 - 10:30

LUGAR: Oficina de Dirección de Carrera de Agrícola.

TEMA: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL.

1. En que consiste el control y monitoreo de las Áreas de producción Agrícola

Existe un asistente o Técnico de Campo al que se le asignan tareas por parte de Dirección y él a través del personal de campo ejecuta las actividades, por lo tanto el mismo técnico controla y monitorea las actividades.

2. ¿De qué manera realizan este proceso y quienes lo realizan?

El proceso es actividad encomendada, actividad terminada, cumplimiento. Lo realiza el Técnico de Campo y Director de Carrera.

3. ¿Presentan alguna inconformidad de llevarlo de esta manera?

Si, solo con este tipo de control no se ha podido llevar un registro estadístico de producción, labores culturales y agronómicas, finalmente ingresos y egresos.

4. ¿Cree usted que el uso de una aplicación informática mejoraría el control y monitoreo de las áreas de producción agrícola?

Por supuesto, no solo por el control, sino además por la planificación de campo.

5. ¿Cuáles son los procesos que realizan para llevar el control y monitoreo de las áreas de producción agrícola de la ESPAM MFL?

Necesidades de los cultivos, épocas de año, actividades emergentes, labores de producción programadas.

6. Tenemos como propuesta elaborar un SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL, ¿Qué opina usted?

Es una necesidad sentida porque nos permitirá planificar, implementar, evaluar y realizar planes de mejora continua.

7. ¿Qué requerimientos consideraría usted que debe tener este Sistema?

- Registro estadístico de información
- Registro de producción por cultivos y épocas de año
- Bitácora de actividades realizadas por cada cultivo, es decir, que se registró, que se usó.
- Planificación y control
- Mapas de uso
- Alertas de corrección
- Puede ser visible para el público
- No existirán más de 8 usuarios, Director, Técnico, Coordinadores de las áreas y estos son por niveles, es decir, el que toma decisiones no lo haría quien hace el operativo.

ANEXO 2

ENTREVISTA AL ING. FELIPE VINCES – PROFESIONAL DE CAMPO

NOMBRE DE ENTREVISTADO: Ing. Felipe Vincés Intriago

CARGO QUE OCUPA EN LA INSTITUCIÓN: Profesional de campo

FECHA Y HORA: 6 de agosto de 2014, 10:00 am

LUGAR: Campus El Limón Área de Cultivo.

TEMA: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL.

1. En que consiste el control y monitoreo de las Áreas de producción Agrícola

Darle mantenimiento tanto en maleza como en fertilización para una mejor producción, control de plagas y enfermedades a las cosechas de los cultivos de cacao y plátano.

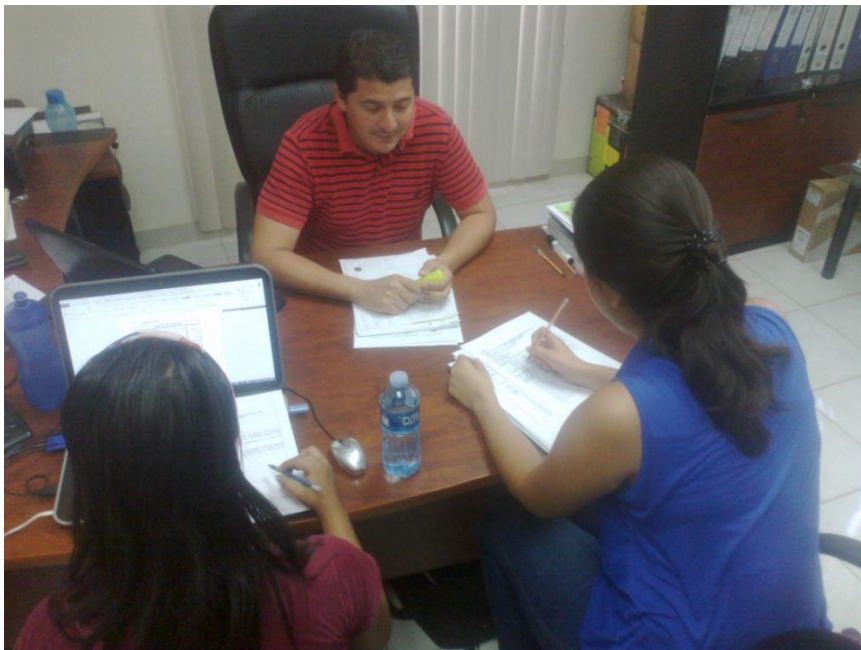
2. ¿De qué manera realizan este proceso y quienes lo realizan?

Lo realizan los trabajadores de forma manual o con los equipos como la motosierra de altura.

3. ¿Cuáles son los procesos que realizan para llevar el control y monitoreo de las áreas de producción agrícola de la ESPAM MFL?

Las labores son como las cosechas manuales.

ANEXO 3
FOTOGRAFÍAS



3 – A. Entrevista al Director de la Carrera Agrícola



3-B. Entrevista realizada el Ing. Lenin Vera Director de la Carrera de Agrícola






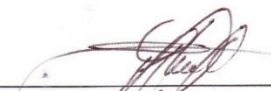
3-C. Entrevista realizada al Técnico de Campo



3-D. Entrevista al Ing. Felipe Vincés, Técnico de campo

ANEXO 4




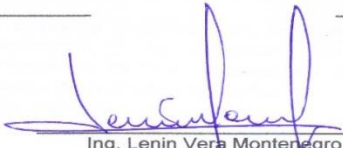
**RESOLUCIONES DE LA OBSERVACIÓN Y ENTREVISTA REALIZADA AL
PROFESIONAL DE CAMPO**

	ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ CARRERA INGENIERÍA EN INFORMÁTICA
ACTA Nº 02/2014	FECHA: 06-08-2014
PERÍODO LECTIVO: ABRIL -SEPTIEMBRE/2014	
PARTICIPANTES: Ing. Felipe Vínces Intriago Srta. Gema Mendoza Srta. Jeniffer Párraga	
Inicia la sesión a las 10h00 am en Campus El Limón Área de Cultivos	
ORDEN DEL DÍA: <ul style="list-style-type: none"> ■ Elaboración de la entrevista al Profesional de Campo y revisión de actividades que se realizan en las áreas de producción agrícola para obtener información correspondiente al desarrollo de Tesis presentados por las postulantes: Mendoza Vera Gema Teresa y Párraga Vera Jeniffer Mercedes <p style="text-align: center;">TÍTULO DE LA TESIS: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL.</p> <ul style="list-style-type: none"> ■ Asuntos Varios ■ Resoluciones 	
PUNTOS TRATADOS: Se inicia la sesión con la constatación de la asistencia de todos los integrantes antes mencionados, para luego proceder a la entrevista y posteriormente a la revisión de información, luego de haber revisado la documentación se procede a emitir la siguiente resolución:	
RESOLUCIÓN: Se determinó que en las Áreas de Producción Agrícola de la ESPAM MFL se realizan actividades como: <ul style="list-style-type: none"> - Mantenimiento en malezas - Fertilización - Control de plagas y enfermedades <p>Estas actividades son realizadas por los trabajadores de forma manual o con equipos de mantenimiento.</p> <p>Termina la sesión a las 10h30.</p>	
 _____ Gema T. Mendoza Vera	 _____ Jeniffer M. Párraga Vera
 _____ Ing. Felipe Vínces Intriago Profesional de Campo	
CC: Tutor de Tesis, Ing. Ángel Vélez Mero	

Acta 1 con fecha 6 de agosto del 2014.

ANEXO 5


RESOLUCIONES DE ACTAS



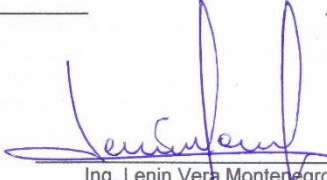
	ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ CARRERA INGENIERÍA EN INFORMÁTICA
ACTA N° 01/2014	FECHA: 31-07-2014
PERÍODO LECTIVO: ABRIL -SEPTIEMBRE/2014	
PARTICIPANTES: Ing. Lenin Vera Montenegro Srta. Gema Mendoza Srta. Jeniffer Párraga	
Inicia la sesión a las 10h30 am en Dirección de la Carrera Agrícola	
ORDEN DEL DÍA:	
<ul style="list-style-type: none"> ■ Elaboración de la entrevista al Director de la Carrera de Agrícola y revisión de los requerimientos funcionales y no funcionales del levantamiento de información correspondiente al desarrollo de Tesis presentados por las postulantes: Mendoza Vera Gema Teresa y Párraga Vera Jeniffer Mercedes <p style="text-align: center;">TÍTULO DE LA TESIS: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL.</p> <ul style="list-style-type: none"> ■ Asuntos Varios ■ Resoluciones 	
PUNTOS TRATADOS: Se inicia la sesión con la constatación de la asistencia de todos los integrantes antes mencionados, para luego proceder a la entrevista y posteriormente a la revisión de los requerimientos del levantamiento de información realizado, luego de haber revisado la documentación se procede a emitir la siguiente resolución:	
RESOLUCIÓN:	
REQUISITOS FUNCIONALES	
<ul style="list-style-type: none"> • Acceder al sistema por medio de Login • Ingreso mediante mapas de uso • Registro de la producción por cultivos • Registro de la producción por épocas del año • Registro estadístico de información • Ingreso de la planificación por cultivos • Ingreso de la planificación por época del año • Control de la planificación de los cultivos • Control de la planificación de cada época del año • Alertas de corrección por cada cultivo • Alertas de corrección por época del año • Registro de actividades por cada cultivo • Reportes semanales por épocas del año y cultivos • Reportes mensuales por épocas del año y cultivos • Reportes estadísticos semanales por épocas del año y cultivos • Reportes estadísticos mensuales por épocas del año y cultivos • Reportes de actividades por cada cultivo • Reportes de alertas de corrección por épocas del año y cultivos 	
REQUERIMIENTOS NO FUNCIONALES	
<ul style="list-style-type: none"> • Seguridad • Efectividad • Confiabilidad • Compatibilidad con diferentes navegadores • El diseño de la base de datos será en SQL Server 2012 • Tendrá una estructura en 3 capas desarrollada en el IDE Visual Studio 2012, haciendo uso del lenguaje de programación C# 	
Termina la sesión a las 11h30.	
 Gema T. Mendoza Vera	 Jeniffer M. Párraga-Vera
 Ing. Lenin Vera Montenegro Director de la Carrera Agrícola	
CC: Tutor de Tesis, Ing. Ángel Vélez Mero	

5-A. Acta 1 con fecha 31 de julio del 2014

 ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ CARRERA INGENIERÍA EN INFORMÁTICA	
ACTA Nº 03/2014	FECHA: 08-10-2014
PERÍODO LECTIVO: OCTUBRE 2014 - MARZO 2015	
PARTICIPANTES: Ing. Lenin Vera Montenegro Srta. Gema Mendoza Srta. Jeniffer Párraga	
Inicia la sesión a las 10h00 am en Campus El Limón Oficina de la Dirección de la Carrera Agrícola	
ORDEN DEL DÍA: <ul style="list-style-type: none"> ■ Definición de nuevos requisitos para la Tesis presentada por las postulantes: Mendoza Vera Gema Teresa y Párraga Vera Jeniffer Mercedes <p style="text-align: center;">TÍTULO DE LA TESIS: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL.</p> <ul style="list-style-type: none"> ■ Asuntos Varios ■ Resoluciones 	
PUNTOS TRATADOS: Se inicia la sesión con la constatación de la asistencia de todos los integrantes antes mencionados, para luego proceder a la definición de nuevos requisitos, luego de haber revisado la documentación se procede a emitir la siguiente resolución:	
RESOLUCIÓN: Se determinó que el Sistema ya no tendrá ingresos por Cultivos y Épocas del año, se lo hará mediante un menú de Labores Agronómicas, este menú tendrá las siguientes opciones: <ul style="list-style-type: none"> - Siembra - Fertilización - Labores Fitosanitarias - Cosechas - Pos-Cosechas - Otros - Informes <p>Cabe recalcar que la opción Informes solo estará habilitada para el Director de Carrera. Así mismo se generaran para cada uno de los menús, es decir, para siembra, fertilización, y uno general que abarque todas las opciones.</p> <p>También se definieron los campos a ingresarse en cada uno de los menús, estos son:</p> <ul style="list-style-type: none"> - Fecha - Nombre de material - Cantidad de material - Unidad de medida del material - Personal - Responsable - Valores Totales <p>Todas las áreas de producción tendrán las mismas opciones de ingreso, y cada coordinador solo tendrá habilitada su área respectiva.</p>	
 _____ Gema T. Mendoza Vera	 _____ Jeniffer M. Párraga Vera
 _____ Ing. Lenin Vera Montenegro Director de la Carrera Agrícola	
CC: Tutor de Tesis, Ing. Ángel Vélez Mero	


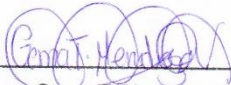


5-B. Acta 2 con fecha 8 de agosto del 2014.

 ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ CARRERA INGENIERÍA EN INFORMÁTICA	
ACTA N° 04/2014	FECHA: 27-11-2014
PERÍODO LECTIVO: OCTUBRE 2014 - MARZO 2015	
PARTICIPANTES: Ing. Lenin Vera Montenegro Srta. Gema Mendoza Srta. Jeniffer Párraga	
Inicia la sesión a las 10h00 am en Campus El Limón Oficina de la Dirección de la Carrera Agrícola	
ORDEN DEL DÍA: <ul style="list-style-type: none"> ■ Establecer nuevos requerimientos de la Tesis presentada por las postulantes: Mendoza Vera Gema Teresa y Párraga Vera Jeniffer Mercedes TÍTULO DE LA TESIS: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL. <ul style="list-style-type: none"> ■ Asuntos Varios ■ Resoluciones 	
PUNTOS TRATADOS: Se inicia la sesión con la constatación de la asistencia de todos los integrantes antes mencionados, para luego proceder a establecer nuevos requisitos del sistema, luego de haber revisado la documentación se procede a emitir la siguiente resolución:	
RESOLUCIÓN: Se determinó que el Sistema tendrá ingreso de Datos Meteorológicos, este se lo realizara diariamente, para poder visualizarlo se lo hará en una pestaña con el Menú de Datos Meteorológicos. Así mismo, se ingresara un Calendario con las fases lunares del año, esto para saber las fechas de poda, de siembra, entre otras. Se agregara un Menú con el nombre de Bodega, aquí se realizara la consulta de los equipos y productos que existen, y tendrá la opción para nuevos ingresos. De los productos y equipos se podrá ingresar y consultar los campos nombre, cantidad, precio y observaciones, es decir, será especie de inventario. Termina la sesión a las 11h00.	

 _____ Gema T. Mendoza Vera	 _____ Jeniffer M. Párraga Vera
 _____ Ing. Lenin Vera Montenegro Director de la Carrera Agrícola	

CC: Tutor de Tesis, Ing. Ángel Vélez Mero

5-C. Acta 4 con fecha 27 de noviembre del 2014.

 ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ CARRERA INGENIERÍA EN INFORMÁTICA	
ACTA Nº 05/2014	FECHA: 22-01-2015
PERÍODO LECTIVO: OCTUBRE 2014 - MARZO 2015	
PARTICIPANTES: Ing. Leonardo Vera Macías Srta. Gema Mendoza Srta. Jeniffer Párraga	
Inicia la sesión a las 10h00 am en Campus El Limón Oficina de Dirección de Carrera Agrícola	
ORDEN DEL DÍA: <ul style="list-style-type: none"> ■ Establecer nuevos requerimientos de la Tesis presentada por las postulantes: Mendoza Vera Gema Teresa y Párraga Vera Jeniffer Mercedes TÍTULO DE LA TESIS: SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESPAM MFL. <ul style="list-style-type: none"> ■ Asuntos Varios ■ Resoluciones 	
PUNTOS TRATADOS: Se inicia la sesión con la constatación de la asistencia de todos los integrantes antes mencionados, para luego proceder a la definición de nuevos requisitos, luego de haber revisado la documentación se procede a emitir la siguiente resolución:	
RESOLUCIÓN: Se determinó que el Sistema solo tendrá 6 tipos de usuarios para las áreas agrícolas, estos son usuario Cacao, que abarca las áreas de cacao clonal, cacao joven, cacao injerto, cacao INIAP y cacao más plátano, Convencional, Orgánico, Frutales y Plátano para cada una de las áreas correspondientes, y usuario <i>Administrador de Campo para las áreas restantes.</i> Así mismo se creara un usuario Meteorológico, este tendrá acceso al ingreso del calendario lunar, datos meteorológicos y algún boletín de un evento o predicciones. Los usuarios Administrador y Director quedan como ya se había establecido anteriormente. Las áreas de Cacao, Convencional, Orgánico, Frutales y Plátano tendrán una bodega cada una, así mismo podrán revisar la bodega general para poder solicitar algún producto a esta, mientras que las áreas que correspondan al usuario Administrador de Campo tendrán únicamente acceso a la bodega principal. Termina la sesión a las 11h00.	
 _____ Gema T. Mendoza Vera	
 _____ Jeniffer M. Párraga Vera	
 _____ Ing. Leonardo Vera Macías Director de Carrea Agrícola	
CC: Tutor de Tesis, Ing. Ángel Vélez Mero	

5-D. Acta 5 con fecha 22 de enero del 2015.

ANEXO 6

CERTIFICACIÓN POR PARTE DEL DIRECTOR DE LA CARRERA DE AGRÍCOLA

REPÚBLICA DEL ECUADOR



ESPAM MFL

ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ
MANUEL FÉLIX LÓPEZ
DIRECCIÓN DE LA CARRERA DE INGENIERÍA AGRÍCOLA



CERTIFICACIÓN

El suscrito Director de la Carrera de Agrícola de la Escuela Superior Politécnica Agropecuaria de Manabí "Manuel Félix López", CERTIFICA:

Que las señoritas: GEMA TERESA MENDOZA VERA portadora de la C.C. 131202044-7 y JENIFFER MERCEDES PÁRRAGA VERA con C.C. 131389416-2, egresadas de la Carrera de Informática, han creado el SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA D LA ESPAM MFL, mismo que ha sido verificado por quien suscribe y cumple con el requerimiento establecido; trabajo realizado previo a la obtención del Título de Ingenieras Informáticas.

Es todo lo que puedo certificar en honor a la verdad, autorizando a los interesados hacer uso del presente documento de la forma que conviniera a sus intereses.

Calceta, 03 de marzo de 2015

Atentamente,

Ing. Leonardo Vera Macías
DIRECTOR DE LA CARRERA DE AGRÍCOLA
C.I. 1309859666
Teléf Ofic: 052 686-101
Celular: 0997744766



LV/an

Campus Politécnico Teléfono: 052-686101 (Ext.114) mail: agricola@espam.edu.ec 1/1

ANEXO 7
MANUAL DE PROGRAMACIÓN

**SISTEMA DE CONTROL Y MONITOREO EN LAS
ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA
ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA
DE MANABÍ MANUEL FÉLIX LÓPEZ**

Manual de programación

El manual tiene como objetivo servir como guía técnica del sistema de control y monitoreo de las áreas de producción agrícola de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López para evitar problemas en futuros cambios y tener una idea ante cualquier tipo de modificación si fuera necesario.

El sistema se desarrolló bajo el lenguaje de programación C# y Visual Studio como Entorno de Desarrollo Integrado o IDE. Se trabajó bajo la programación orientada a objetos y se dividió la solución del sistema en tres capas.

CAPA DE ACCESO A DATOS

En la capa de acceso a datos se hizo una conexión por medio de dataset y al mismo tiempo se hizo una segunda conexión a través de clases.

Se creó una clase llamada AD_CI_Cadena.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;

namespace C_Datos
{
    public class AD_CI_Cadena
    {
        public static SqlConnection ObtnerConexion()
        {
            SqlConnection Conn = new SqlConnection("Data source=.; Initial
            Catalog=bd_areasagricolas_espam; Integrated Security=True");
            Conn.Open();
            return Conn;
        }
    }
}
```

Una segunda clase llamada CI_Usuario

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Data;
using System.Data.SqlClient;
using C_Datos;
namespace C_Datos
{
    public class Cl_Usuario
    {
        public static int Login(string Nom_usuario, string Contraseña)
        {
            //var cn = new SqlConnection(AD_Cl_Cadena.ObtnerConexion);
            SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();
            using (conexion)
            {
                var consulta = "sp_Ingresar";
                var cmd = new SqlCommand(consulta, conexion);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@Nom_usuario", Nom_usuario);
                cmd.Parameters.AddWithValue("@Contraseña", Contraseña);
                //conexion.Close();
                return Convert.ToInt32(cmd.ExecuteScalar());
            }
        }

        public static string TipoIngreso(string Nom_usuario, string Contraseña)
        {
            SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();
            using (conexion)
            {
                var consulta = "sp_TipoIngreso";
                var cmd = new SqlCommand(consulta, conexion);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@nick", Nom_usuario);
                cmd.Parameters.AddWithValue("@cla", Contraseña);
                //conexion.Open();
                return Convert.ToString(cmd.ExecuteScalar());
            }
        }
    }
}

```

CAPA DE NEGOCIOS

En la capa de negocios se hizo uso de librerías de clases, en el lenguaje de programación C# y bajo programación orientada a objetos, se debió considerar agregar las referencias de la capa de datos para hacer posible la gestión con la base de datos. Se crearon clases como:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using C_Datos.DataSet1TableAdapters;

namespace C_Negocio
{
    public class Cl_Areas
    {
        public int id_areas { get; set; }
        public string Nombres_areas { get; set; }

        public List<Cl_Areas> met_ObtenerAreas()
        {
            List<Cl_Areas> lista = new List<Cl_Areas>();
            sp_ConsutaAreasTableAdapter tabla = new
            sp_ConsutaAreasTableAdapter();

            foreach (var item in tabla.GetData().ToList())
            {
                lista.Add(new Cl_Areas { id_areas = item.id_areasagricolas,
                Nombres_areas = item.Nombre_areas });
            }

            return lista;
        }

        public string met_ingresarareas(Cl_Areas area)
        {
            string mensaje = "Datos ingresados correctamente";
            try
            {
                QueriesTableAdapter tabla = new QueriesTableAdapter();
                tabla.sp_InsertarAreas(area.Nombres_areas);
            }
            catch (Exception ex)
            {
                mensaje = ex.Message;
            }
            return mensaje;
        }
    }
}

```

Y así se fueron creando clases para cada una de las tablas de la base de datos, de la misma manera se realizaron clases necesarias para la inserción, modificación y consulta de registros.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using C_Datos.DataSet1TableAdapters;

```

```

using System.Data.SqlClient;
using C_Datos;

namespace C_Negocio
{
    public class Cl_CacaoClonal
    {
        public int id_campos { get; set; }
        public int id_labores { get; set; }
        public DateTime Fecha { get; set; }
        public string Responsables { get; set; }
        public decimal Valores_Totales { get; set; }
        public int id_areasagricolas { get; set; }
        public int id_Producto { get; set; }
        public string Observacion { get; set; }

        public static List<CacaoClonal> met_CacaoClonalSiembra(int id_lab,
DateTime fech, DateTime fech1)
        {
            List<CacaoClonal> lista = new List<CacaoClonal>();
            SqlConnection conn = AD_Cl_Cadena.ObtnerConexion();
            {
                string query = @"SELECT          tb_campos.id_campos,
tb_campos.id_labores, tb_campos.Fecha, tb_campos.Responsables,
tb_campos.Valores_Totales, tb_campos.id_areasagricolas, tb_campos.id_producto,
tb_campos.Observacion, tb_producto.nom_producto as Producto
FROM          tb_campos INNER
JOIN
tb_producto ON
tb_campos.id_producto = tb_producto.id_producto
INNER JOIN
tb_labores ON
tb_labores.id_labores = tb_labores.id_labores

WHERE          (tb_campos.id_areasagricolas = 13) and (tb_labores.id_labores
= tb_campos.id_labores) and (tb_labores.id_labores = @id_labores ) and
tb_campos.Fecha between @Fecha and @Fecha1";

                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@id_labores", id_lab);
                cmd.Parameters.AddWithValue("@Fecha", fech);
                cmd.Parameters.AddWithValue("@Fecha1", fech1);
                SqlDataReader reader = cmd.ExecuteReader();
                while (reader.Read())
                {
                    CacaoClonal img = new
CacaoClonal(/*Convert.ToInt32(reader["id_campos"]),
Convert.ToInt32(reader["id_labores"]),*/ Convert.ToDateTime(reader["Fecha"]),
Convert.ToString(reader["Responsables"]),
Convert.ToDecimal(reader["Valores_Totales"]),
/*Convert.ToInt32(reader["id_areasagricolas"]),
Convert.ToInt32(reader["id_Producto"]),*/ Convert.ToString(reader["Producto"]),
Convert.ToString(reader["Observacion"]));
                    lista.Add(img);
                }
            }
            return lista;
        }
    }
}

```

```

    }

    public class CacaoClonal
    {
        public CacaoClonal(/*int id_campos, int id_labores,*/ DateTime fecha,
string responsable, decimal Valores_Totales,/*int id_areasagricolas, int
id_producto,*/ string Producto, string Observacion)
        {

            //this.id_campos = id_campos;
            //this.id_labores = id_labores;
            this.Fecha = fecha;
            this.Responsables = responsable;
            this.Valores_Totales = Valores_Totales;
            //this.id_areasagricolas = id_areasagricolas;
            //this.id_Producto = id_Producto;
            this.Producto = Producto;
            this.Observacion = Observacion;

        }

        public CacaoClonal() { }

        //public int id_campos { get; set; }
        //public int id_labores { get; set; }
        public DateTime Fecha { get; set; }
        public string Responsables { get; set; }
        public decimal Valores_Totales { get; set; }
        //public int id_areasagricolas { get; set; }
        //public int id_Producto { get; set; }
        public string Producto { get; set; }
        public string Observacion { get; set; }

    }

    public List<Cl_CacaoClonal> cacaoClonalSiembra()
    {
        List<Cl_CacaoClonal> lista = new List<Cl_CacaoClonal>();
        sp_CacaoClonalSiembraTableAdapter dato = new
sp_CacaoClonalSiembraTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoClonal { id_campos = item.id_campos,
id_labores = item.id_labores, Fecha = item.Fecha, Responsables =
item.Responsables, Valores_Totales = item.Valores_Totales, id_areasagricolas =
item.id_areasagricolas, id_Producto = item.id_producto, Observacion =
item.Observacion });
        }
        return lista;
    }

    public List<Cl_CacaoClonal> cacaoClonalFertilizacion()
    {
        List<Cl_CacaoClonal> lista = new List<Cl_CacaoClonal>();
        sp_CacaoClonalFertilizacionTableAdapter dato = new
sp_CacaoClonalFertilizacionTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {

```



```

        lista.Add(new Cl_CacaoClonal { id_campos = item.id_campos,
id_labores = item.id_labores, Fecha = item.Fecha, Responsables =
item.Responsables, Valores_Totales = item.Valores_Totales, id_areasagricolas =
item.id_areasagricolas, id_Producto = item.id_producto, Observacion =
item.Observacion });
    }
    return lista;
}

public List<Cl_CacaoClonal> cacaoClonalLaboresFitosanitarias()
{
    List<Cl_CacaoClonal> lista = new List<Cl_CacaoClonal>();
    sp_CacaoClonarLaboresFitosanitariasTableAdapter dato = new
sp_CacaoClonarLaboresFitosanitariasTableAdapter();
    foreach (var item in dato.GetData().ToList())
    {
        lista.Add(new Cl_CacaoClonal { id_campos = item.id_campos,
id_labores = item.id_labores, Fecha = item.Fecha, Responsables =
item.Responsables, Valores_Totales = item.Valores_Totales, id_areasagricolas =
item.id_areasagricolas, id_Producto = item.id_producto, Observacion =
item.Observacion });
    }
    return lista;
}

public List<Cl_CacaoClonal> cacaoClonalCosecha()
{
    List<Cl_CacaoClonal> lista = new List<Cl_CacaoClonal>();
    sp_CacaoClonalCosechaTableAdapter dato = new
sp_CacaoClonalCosechaTableAdapter();
    foreach (var item in dato.GetData().ToList())
    {
        lista.Add(new Cl_CacaoClonal { id_campos = item.id_campos,
id_labores = item.id_labores, Fecha = item.Fecha, Responsables =
item.Responsables, Valores_Totales = item.Valores_Totales, id_areasagricolas =
item.id_areasagricolas, id_Producto = item.id_producto, Observacion =
item.Observacion });
    }
    return lista;
}

public List<Cl_CacaoClonal> cacaoClonalPostCosecha()
{
    List<Cl_CacaoClonal> lista = new List<Cl_CacaoClonal>();
    sp_CacaoClonalPostCosechaTableAdapter dato = new
sp_CacaoClonalPostCosechaTableAdapter();
    foreach (var item in dato.GetData().ToList())
    {
        lista.Add(new Cl_CacaoClonal { id_campos = item.id_campos,
id_labores = item.id_labores, Fecha = item.Fecha, Responsables =
item.Responsables, Valores_Totales = item.Valores_Totales, id_areasagricolas =
item.id_areasagricolas, id_Producto = item.id_producto, Observacion =
item.Observacion });
    }
    return lista;
}

public List<Cl_CacaoClonal> cacaoClonalOtros()
{
    List<Cl_CacaoClonal> lista = new List<Cl_CacaoClonal>();

```

```

        sp_CacaoClonalOtrosTableAdapter dato = new
sp_CacaoClonalOtrosTableAdapter();
        foreach (var item in dato.GetData().ToList())
        {
            lista.Add(new Cl_CacaoClonal { id_campos = item.id_campos,
id_labores = item.id_labores, Fecha = item.Fecha, Responsables =
item.Responsables, Valores_Totales = item.Valores_Totales, id_areasagricolas =
item.id_areasagricolas, id_Producto = item.id_producto, Observacion =
item.Observacion });
        }
        return lista;
    }
}
}
}

```

Clase Cargar Archivos

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using C_Datos.DataSet1TableAdapters;
using System.Data.SqlClient;
using C_Datos;

namespace C_Negocio
{
    public class Cl_CargarArchivo
    {
        public static void GuardarDocumento(string nombre, string descripcion,
DateTime fecha, byte[] archivo)
        {
            SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();

            SqlCommand cmd = new SqlCommand("sp_GuardarArchivo", conexion);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@nombre", nombre);
            cmd.Parameters.AddWithValue("@descripcion", descripcion);
            cmd.Parameters.AddWithValue("@fecha", fecha);
            SqlParameter archParam = cmd.Parameters.Add("@archivo",
System.Data.SqlDbType.VarBinary);
            archParam.Value = archivo;
            cmd.ExecuteNonQuery();
        }

        public static void GuardarImagen(string nombre, byte[] archivo)
        {
            SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();

            SqlCommand cmd = new SqlCommand("sp_GuardarImagen", conexion);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@nombre", nombre);

```

```

        SqlParameter archParam = cmd.Parameters.Add("@imagen",
System.Data.SqlDbType.Image);
        archParam.Value = archivo;
        cmd.ExecuteNonQuery();

    }

    public class Archivo
    {
        public Archivo(int id_archivo, string nombre, string descripcion,
DateTime fecha)
        {
            this.id_archivo = id_archivo;
            this.Nombre = nombre;
            this.Descripcion = descripcion;
            this.Fecha = fecha;
        }

        public Archivo() { }

        public int id_archivo { get; set; }
        public string Nombre { get; set; }
        public string Descripcion { get; set; }
        public DateTime Fecha { get; set; }
        public byte[] Archivos { get; set; }
    }

    public static Archivo GetById(int Id)
    {
        Archivo arch = null;
        SqlConnection conn = BDComun.ObtnerConexion();
        {
            string query = @"top 10 SELECT id_archivo, Nombre, Descripcion,
Fecha, Archivo
                                FROM          tb_archivo
                                WHERE id_archivo = @id_archivo
                                ORDER BY id_archivo DESC";

            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@id_archivo", Id);

            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                arch = new Archivo(Convert.ToInt32(reader["id_archivo"]),
Convert.ToString(reader["Nombre"]), Convert.ToString(reader["Descripcion"]),
Convert.ToDateTime(reader["Fecha"]));
                arch.Archivos = (byte[])reader["Archivo"];
            }
        }
        return arch;
    }

    public static List<Archivo> GetAll()
    {
        List<Archivo> lista = new List<Archivo>();
        SqlConnection conn = BDComun.ObtnerConexion();
        {
            string query = @"SELECT id_archivo,Nombre, Descripcion, Fecha

```

```

        FROM          tb_archivo";
        SqlCommand cmd = new SqlCommand(query, conn);
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            Archivo img = new
Archivo(Convert.ToInt32(reader["id_archivo"]),
Convert.ToString(reader["nombre"]), Convert.ToString(reader["descripcion"]),
Convert.ToDateTime(reader["fecha"]));
            lista.Add(img);
        }
    }
    return lista;
}

public class Imagen
{
    public Imagen(int id_image, string nombre)
    {
        this.id_imagen = id_image;
        this.Nombre = nombre;
    }

    public Imagen() { }

    public int id_imagen { get; set; }
    public string Nombre { get; set; }
    public byte[] Images { get; set; }
}

public static Imagen ObtenerById(int id_imagen)
{
    Imagen imagenes = null;
    SqlConnection conn = BDComun.ObtnerConexion();
    {
        string query = @"SELECT id_imagen,Nombre,Imagen
                        FROM tb_imagen WHERE id_imagen = @id_imagen";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@id_imagen", id_imagen);
        SqlDataReader reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            imagenes = ConvertirImagen(reader);
        }
        conn.Close();
    }
    return imagenes;
}

public static List<Imagen> ObtenerTodos()
{
    List<Imagen> lista = new List<Imagen>();
    SqlConnection conn = BDComun.ObtnerConexion();
    {
        string query = @"SELECT id_imagen, Nombre, Imagen
                        FROM tb_imagen";

        SqlCommand cmd = new SqlCommand(query, conn);
        SqlDataReader reader = cmd.ExecuteReader();
    }
}

```

```

        while (reader.Read())
        {
            lista.Add(ConvertirImagen(reader));
        }
        conn.Close();
    }

    return lista;
}
private static Imagen ConvertirImagen(IDataReader reader)
{
    Imagen imagens = new Imagen();
    imagens.id_imagen = Convert.ToInt32(reader["id_imagen"]);
    imagens.Nombre = Convert.ToString(reader["Nombre"]);
    if (reader["Imagen"] != DBNull.Value)
        imagens.Images = (byte[])reader["Imagen"];

    return imagens;
}

public static Imagen Actualizar(Imagen imagin)
{
    SqlConnection conn = BDComun.ObtnerConexion();
    {
        string query = @"UPDATE tb_imagen SET
                        Nombre = @Nombre,
                        Imagen = ISNULL(@Imagen, Imagen)
                        WHERE id_imagen = ( select max(id_imagen) from
tb_imagen)";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Nombre", imagin.Nombre);
        SqlParameter imageParam = cmd.Parameters.Add("@Imagen",
System.Data.SqlDbType.Image);

        cmd.ExecuteNonQuery();
    }

    return imagin;
}

public static void ActualizarImagen(string nombre, byte[] imagen)
{
    SqlConnection conexion = AD_Cl_Cadena.ObtnerConexion();

    SqlCommand cmd = new SqlCommand("act", conexion);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@nombre", nombre);
    SqlParameter archParam = cmd.Parameters.Add("@imagen",
System.Data.SqlDbType.Image);
    archParam.Value = imagen;
    cmd.ExecuteNonQuery();

}

}
}
}

```

CAPA PRESENTACIÓN

En la capa presentación se crearon varios formularios para cada una de las labores que se practican en las áreas de producción agrícola.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using C_Negocio;
using System.Text;

namespace Presentacion.areacacaoclonal13
{
    public partial class cosechasacc : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                llenarCombo();
            }

            Literal1.Text = CargarDatos();
        }

        public string CargarDatos()
        {
            Cl_CacaoClonal dato = new Cl_CacaoClonal();
            StringBuilder tabla = new StringBuilder();

            tabla.Append("<table artista='tb_campos' class='tb_campos'
rules='all'><thead><tr>");
            tabla.Append("<th>Id</th>");
            tabla.Append("<th>Id Labores</th>");
            tabla.Append("<th>Fecha</th>");
            tabla.Append("<th>Responsables</th>");
            tabla.Append("<th>Valores Totales</th>");
            tabla.Append("<th>Id Areas Agricolas</th>");
            tabla.Append("<th>Id Producto</th>");
            tabla.Append("<th>Observacion</th>");
            tabla.Append("</thead></tr>");
            tabla.Append("<tbody>");

            foreach (var item in dato.cacaoClonalCosecha().ToList())
            {
                tabla.Append("<tr>");

                tabla.Append("<td><a href='/F_Modificar.aspx?id=" +
item.id_campos + "'>Modificar</a><br />");

                tabla.Append("<td>" + item.id_labores + "</td>");
                tabla.Append("<td>" + item.Fecha + "</td>");
            }
        }
    }
}

```

```

        tabla.Append("<td>" + item.Responsables + "</td>");
        tabla.Append("<td>" + item.Valores_Totales + "</td>");
        tabla.Append("<td>" + item.id_areasagricolas + "</td>");
        tabla.Append("<td>" + item.id_Producto + "</td>");
        tabla.Append("<td>" + item.Observacion + "</td>");
        tabla.Append("<tr>");
    }

    tabla.Append("</tbody>");
    tabla.Append("</table>");

    return tabla.ToString();
}

public void llenarCombo()
{
    Cl_Producto com = new Cl_Producto();
    DropDownList1.DataSource = com.met_ObtenerProducto();
    DropDownList1.DataTextField = "Nombre_Producto";
    DropDownList1.DataValueField = "id_Producto";
    DropDownList1.DataBind();
}

protected void Button1_Click(object sender, EventArgs e)
{
    Response.Redirect("../F_Areas/areacacaoclonal.aspx");
}

void almacenardatos()
{
    Cl_Campos objeto = new Cl_Campos();
    Label5.Text = objeto.met_ingresarCampos(new Cl_Campos { id_labores =
4, Fecha = Convert.ToDateTime(TextBox2.Text), Responsables = TextBox3.Text,
Valores_Totales = Convert.ToDecimal(TextBox4.Text), id_areasagricolas = 13,
id_Producto = Convert.ToInt32(DropDownList1.SelectedValue ), Observaciones =
TextBox5.Text });
}

protected void Button2_Click1(object sender, EventArgs e)
{
    almacenardatos();

    limpiar();
}

public void limpiar()
{
    this.TextBox2.Text = " ";
    this.TextBox3.Text = " ";
    TextBox4.Text = " ";
}
}
}

```

Así mismo cada área de producción tiene su formulario principal.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using C_Negocio;
using System.Text;

namespace Presentacion.F_Areas
{
    public partial class areaareaorganica : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

            Literal1.Text = CargarDatos();
        }
        public string CargarDatos()
        {
            Cl_Producto dato = new Cl_Producto();
            StringBuilder tabla = new StringBuilder();

            tabla.Append("<table artista='tb_campos' class='tb_campos'
rules='all'><thead><tr>");
            tabla.Append("<th>Nombre</th>");
            tabla.Append("<th>Hectareas</th>");
            tabla.Append("<th>Epoca de Año</th>");
            tabla.Append("<th>Fecha de Registro</th>");
            tabla.Append("</thead></tr>");
            tabla.Append("<tbody>");

            foreach (var item in dato.met_ObtenerProducto().ToList())
            {
                tabla.Append("<tr>");

                tabla.Append("<td>" + item.Nombre_Producto + "</td>");

                tabla.Append("<td>" + item.Hectareas + "</td>");
                tabla.Append("<td>" + item.EpocaAnio + "</td>");
                tabla.Append("<td>" + item.FechaRegistro + "</td>");
                tabla.Append("<tr>");
            }

            tabla.Append("</tbody>");
            tabla.Append("</table>");

            return tabla.ToString();
        }

        protected void Button1_Click(object sender, EventArgs e)
        {

```



```

        Response.Redirect("../areaorganica9/siembraao.aspx");
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("../areaorganica9/fertilizacionao.aspx");
    }

    protected void Button3_Click(object sender, EventArgs e)
    {
        Response.Redirect("../areaorganica9/laboresfitosanitariasao.aspx");
    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        Response.Redirect("../areaorganica9/cosechasao.aspx");
    }

    protected void Button5_Click(object sender, EventArgs e)
    {
        Response.Redirect("../areaorganica9/postcosechasao.aspx");
    }

    protected void Button6_Click(object sender, EventArgs e)
    {
        Response.Redirect("../areaorganica9/otrosao.aspx");
    }

    protected void Button7_Click(object sender, EventArgs e)
    {
        Response.Redirect("../areaorganica9/informesao.aspx");
    }

    protected void Button8_Click(object sender, EventArgs e)
    {
        almacenardatos();
        limpiar();
    }

    void limpiar()
    {
        TextBox1.Text = " ";
        TextBox2.Text = " ";
        TextBox3.Text = " ";
        TextBox4.Text = " ";
    }

    void almacenardatos()
    {
        Cl_Producto ingreso = new Cl_Producto();
        Label5.Text = ingreso.met_IngresoProducto(new Cl_Producto {
Nombre_Producto = TextBox2.Text, Hectareas = Convert.ToDecimal(TextBox3.Text),
FechaRegistro = Convert.ToDateTime(TextBox1.Text), EpocaAnio = TextBox4.Text });
    }

    protected void Button9_Click(object sender, EventArgs e)
    {
        Response.Redirect("../BodegaOrgnica.aspx");
    }

```

```

protected void Button11_Click(object sender, EventArgs e)
{
    string _open =
"window.open('../areaorganica9/informesareaorganica.aspx', '_newtab');"
    ScriptManager.RegisterStartupScript(this, this.GetType(),
Guid.NewGuid().ToString(), _open, true);
}
}
}

```

De la misma manera existe el formulario para cargar el calendario lunar y archivos de información meteorológica.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using C_Negocio;

namespace Presentacion.F_Meteorologia
{
    public partial class Cargarimagenydocumentos : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            if (FileUpload1.HasFile)
            {
                using (BinaryReader reader = new
BinaryReader(FileUpload1.PostedFile.InputStream))
                {
                    string ext =
System.IO.Path.GetExtension(this.FileUpload1.PostedFile.FileName);
                    if (ext == ".jpg" || ext == ".jpeg" || ext == ".gif" || ext
== ".png" || ext == ".JPG" || ext == ".JPEG" || ext == ".GIF" || ext == ".PNG")
                    {
                        byte[] image =
reader.ReadBytes(FileUpload1.PostedFile.ContentLength);
                        Cl_CargarArchivo.GuardarImagen(FileUpload1.FileName, image);
                        Button4.Visible = true;
                        Button2.Visible = false;
                        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Imagen Cargada Correctamente" +
"');</script>");
                        limpiar();
                    }
                    else
                {

```

```

        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + " Formato de archivo incorrecto. Solo se
permiten imágenes con formato .jpg -- .jpeg -- .gif -- .png" + "');</script>");
    }
}
else
{
    this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Todos lo campos son obligatorios" +
"');</script>");
}
}

protected void Button3_Click(object sender, EventArgs e)
{
    GuardarPdf();
}

private void GuardarPdf()
{
    if (FileUpload2.HasFile && TextBox2.Text != "")
    {
        using (BinaryReader reader = new
BinaryReader(FileUpload2.PostedFile.InputStream))
        {
            string ext =
System.IO.Path.GetExtension(this.FileUpload2.PostedFile.FileName);
            if (ext == ".pdf" )
            {
                byte[] image =
reader.ReadBytes(FileUpload1.PostedFile.ContentLength);
                Cl_CargarArchivo.GuardarDocumento(FileUpload2.FileName,
TextBox2.Text, Convert.ToDateTime(System.DateTime.Now), image);
                this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Archivo cargado correctamente" +
"');</script>");
                limpiar();
            }
            else
            {
                this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + " Formato de archivo incorrecto. Por
favor ingrese un archivo con formato .pdf" + "');</script>");
            }
        }
    }
    else
    {
        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Todos lo campos son obligatorios" +
"');</script>");
        limpiar();
    }
}

private void limpiar()
{
    TextBox2.Text = string.Empty;
}
}

```

```

protected void Button4_Click(object sender, EventArgs e)
{
    ActualizarImagen();
}
private void ActualizarImagen()
{
    if (FileUpload1.HasFile)
    {
        using (BinaryReader reader = new
BinaryReader(FileUpload1.PostedFile.InputStream))
        {
            string ext =
System.IO.Path.GetExtension(this.FileUpload1.PostedFile.FileName);
            if (ext == ".jpg" || ext == ".jpeg" || ext == ".gif" || ext
== ".png" || ext == ".JPG" || ext == ".JPEG" || ext == ".GIF" || ext == ".PNG")
            {
                byte[] image =
reader.ReadBytes(FileUpload1.PostedFile.ContentLength);
                Cl_CargarArchivo.ActualizarImagen(FileUpload1.FileName,
image);
                this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Imagen actualizada correctamente" +
"');</script>");
                limpiar();
            }
            else
            {
                this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + " Formato de archivo incorrecto. Solo se
permiten imágenes con formato .jpg -- .jpeg -- .gif -- .png" + "');</script>");
            }
        }
    }
    else
    {
        this.Page.Response.Write("<script
language='JavaScript'>window.alert('" + "Todos lo campos son obligatorios" +
"');</script>");
        limpiar();
    }
}

protected void Button5_Click(object sender, EventArgs e)
{
    Response.Redirect("/IndexMeteorologia.aspx");
}
}
}

```

ANEXO 8
MANUAL DE USUARIO

**SISTEMA DE CONTROL Y MONITOREO EN LAS
ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA
ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA
DE MANABÍ MANUEL FÉLIX LÓPEZ**

Manual de usuario

SISTEMA DE CONTROL Y MONITOREO EN LAS ÁREAS DE PRODUCCIÓN AGRÍCOLA DE LA ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ

Manual de usuario

Este sistema propone llevar un control de las diversas áreas de producción que tiene la Carrera de Agrícola, entre ellas están el registro de las siembras, cosechas, postcosechas, labores fitosanitarias, fertilización y otros. Así mismo permite el registro de bodega tanto General como de las áreas Cacao, Frutales, Convencional, Orgánica y Plátano, además el ingreso de Datos Meteorológicos y boletines relacionados con los mismos. Este documento servirá de guía para la utilización del sistema.

ÍNDICE

MANUAL DE USUARIO	115
1. Ingreso al sistema	115
2. Selección del área agrícola a través de un mapa	116
3. Selección de labores agronómicas.....	118
4. Ingreso de labores agronómicas	119
5. Uso para administrador	121
6. Datos meteorológicos	125
7. Generar Informes.....	127
8. Contactos.....	128

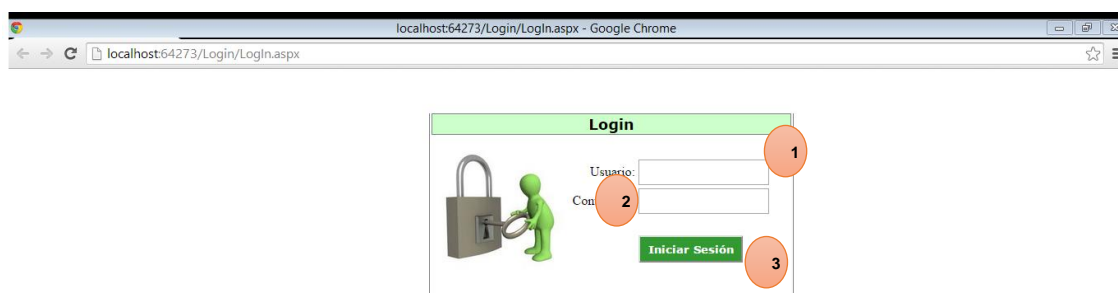
MANUAL DE USUARIO

1. Ingreso al sistema

1.1. Ingreso de usuario

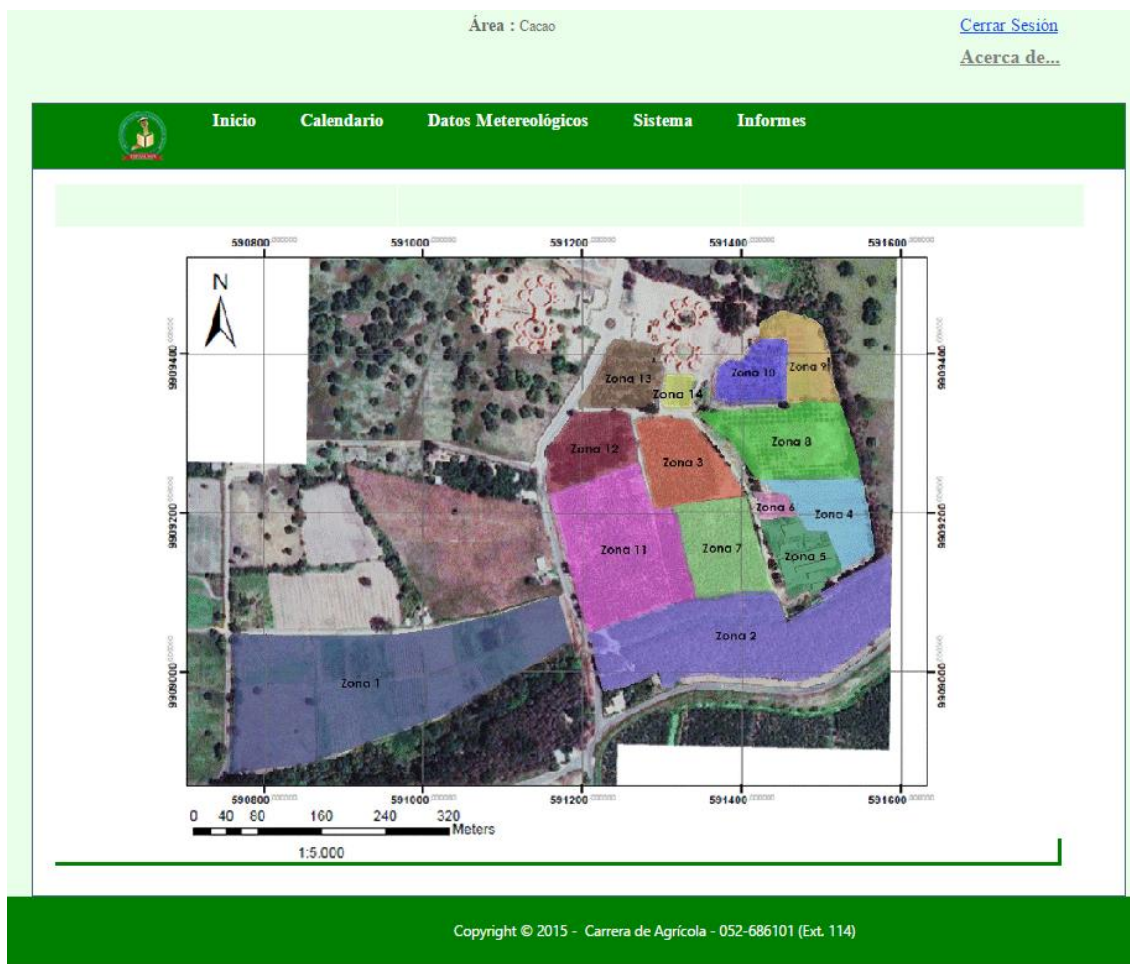
1.2. Ingreso de contraseña

1.3. Botón para el inicio de sesión

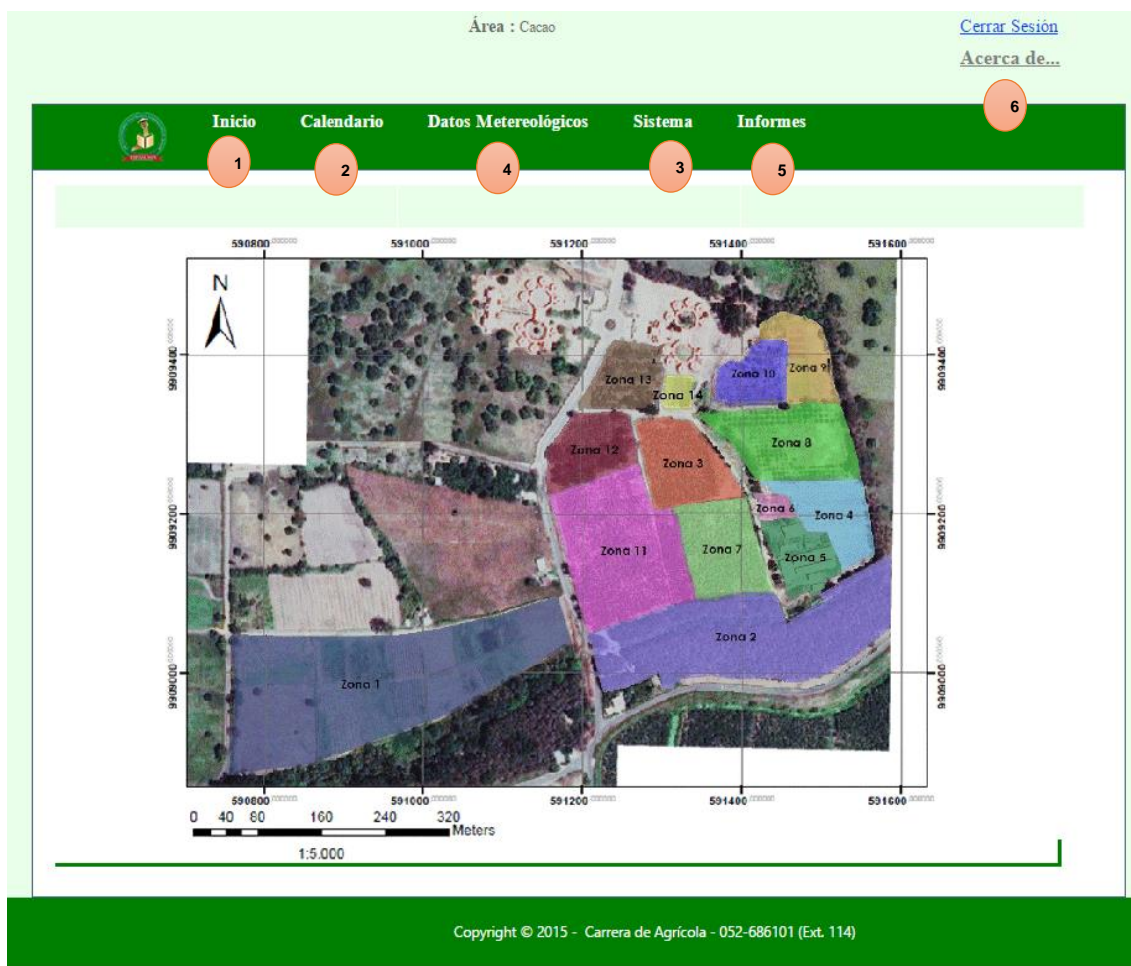


- 1) **INGRESO DE USUARIO:** En este cuadro se debe ingresar el nombre de usuario.
- 2) **INGRESO DE CONTRASEÑA:** En este cuadro se ingresa la contraseña para el respectivo ingreso.
- 3) **BOTÓN DE INICIO DE SESIÓN:** Una vez ingresado el usuario y contraseña dar clic en el botón inicio de sesión y se ingresa al sistema.

2. Selección del área agrícola a través de un mapa



Esta ventana muestra un mapa donde se encuentran localizadas cada una de las áreas de producción agrícola, en el cual basta dar un clic sobre el área correspondiente y automáticamente abrirá una nueva ventana para seleccionar las opciones necesarias.



- 1) MENÚ INICIO: Direcciona a la ventana inicio.
- 2) MENÚ CALENDARIO: Direcciona al calendario agrícola y boletines oficiales.
- 3) ADATOS METEREOLÓGICOS: Direcciona a una ventana donde se puede consultar los datos metereológicos del día.
- 4) MENÚ SISTEMA: Muestra el mapa de las áreas de producción agrícola.
- 5) INFORMES: Muestra los informes de las áreas de producción y cultivos.
- 6) ACERCA DE: Muestra una breve descripción de las áreas de producción.

3. Selección de labores agronómicas

Área : Cacao [Cerrar Sesión](#)
[Acerca de...](#)

Inicio Calendario Datos Metereológicos Sistema Informes

CARRERA DE AGRÍCOLA

Sistema de Control y Monitoreo de Las Areas de Producción Agrícola

Zona 13: Cacao Clonal

Labores Agronómicas

 1 Siembra	 2 Fertilización	 3 Labores Fitos.
 4 Cosechas	 5 Post-Cosechas	 6 Otros

Copyright © 2015 - Carrera de Agrícola - 052-686101 (Ext. 114)

Pantalla que permite seleccionar la labor agronómica

- 1) BOTÓN SIEMBRA: Direcciona a la ventana de siembra.
- 2) BOTÓN FERTILIZACIÓN: Direcciona a la ventana fertilización.
- 3) BOTÓN LABORES FITOSANITARIAS: Direcciona a la ventana labores fitosanitarias.
- 4) BOTÓN COSECHA: Direcciona a la ventana cosecha.
- 5) BOTÓN POST COSECHA: Direcciona a la ventana post cosecha.
- 6) BOTÓN OTROS: Direcciona a la ventada de otros.

4. Ingreso de labores agronómicas

Área : Cacao [Cerrar Sesión](#)
[Acerca de...](#)

Inicio **Calendario** **Datos Metereológicos** **Sistema** **Informes**

CARRERA DE AGRÍCOLA
Sistema de Control y Monitoreo de Las Areas de Producción Agrícola

Zona 13: Cacao Clonal

Labor Agronómica: Siembra



1 Fecha:

2 Responsable: 6

3 Valor Total:

4 Cultivo: 7

5 Observación:

8

Id	Fecha	Responsable	Valores Totales	Observación
Modificar	01/07/2015 0:00:00	Jeniffer Parraga	47,00	htre
Modificar	01/07/2015 0:00:00	Jeniffer Parraga	47,00	htre
Modificar	08/07/2015 0:00:00	Jeniffer Parraga	47,00	htre

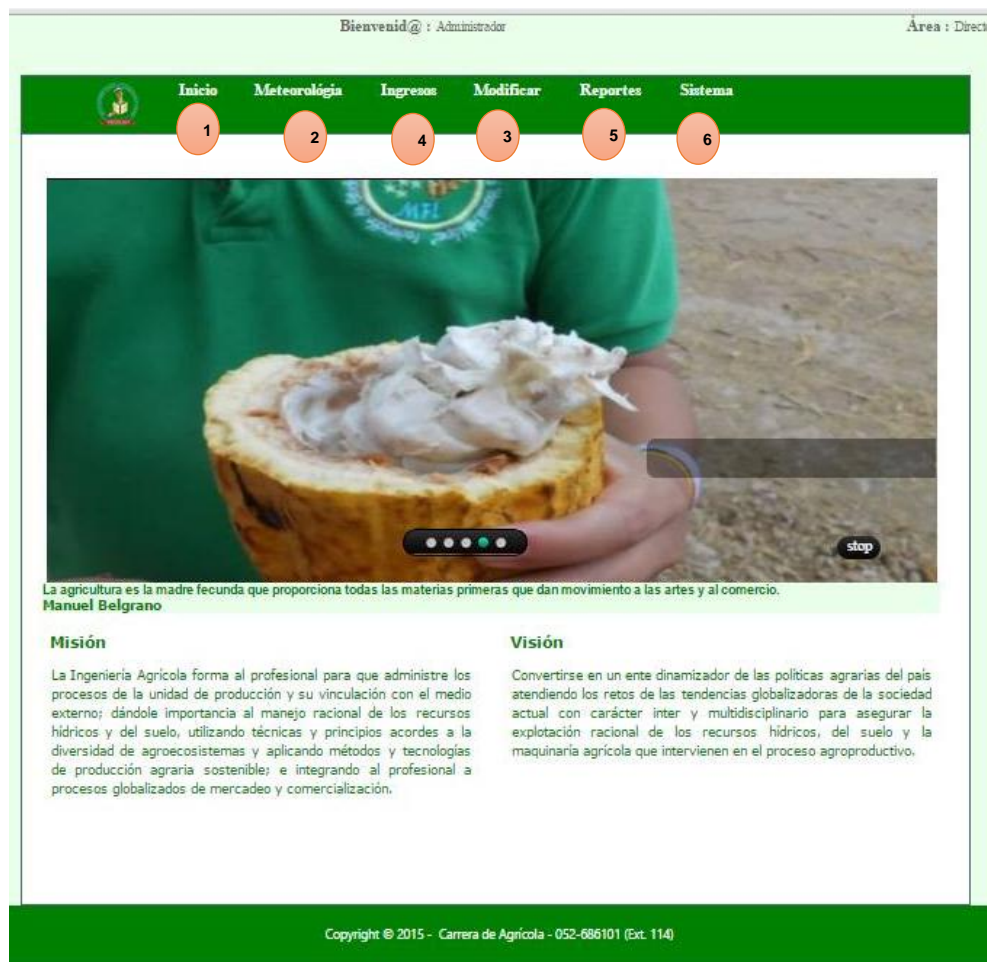
Copyright © 2015 - Carrera de Agrícola - 052-686101 (Ext. 114)

Pantalla de ingreso de la labor siembra.

- 1) INGRESO DE FECHA: Permite ingresar la fecha de cada labor.
- 2) SELECCIÓN DE RESPONSABLE: Permite seleccionar el responsable de la labor registrada.
- 3) INGRESO DE VALORES TOTALES: Permite el ingreso del valor de la labor.
- 4) SELECCIÓN DE PRODUCTO: Selecciona el nombre del producto con el que se trabajó.
- 5) OBSERVACIONES: Ingresar una descripción de actividad a registrar.
- 6) BOTON AGREGAR: redirecciona al formulario de ingresar responsable

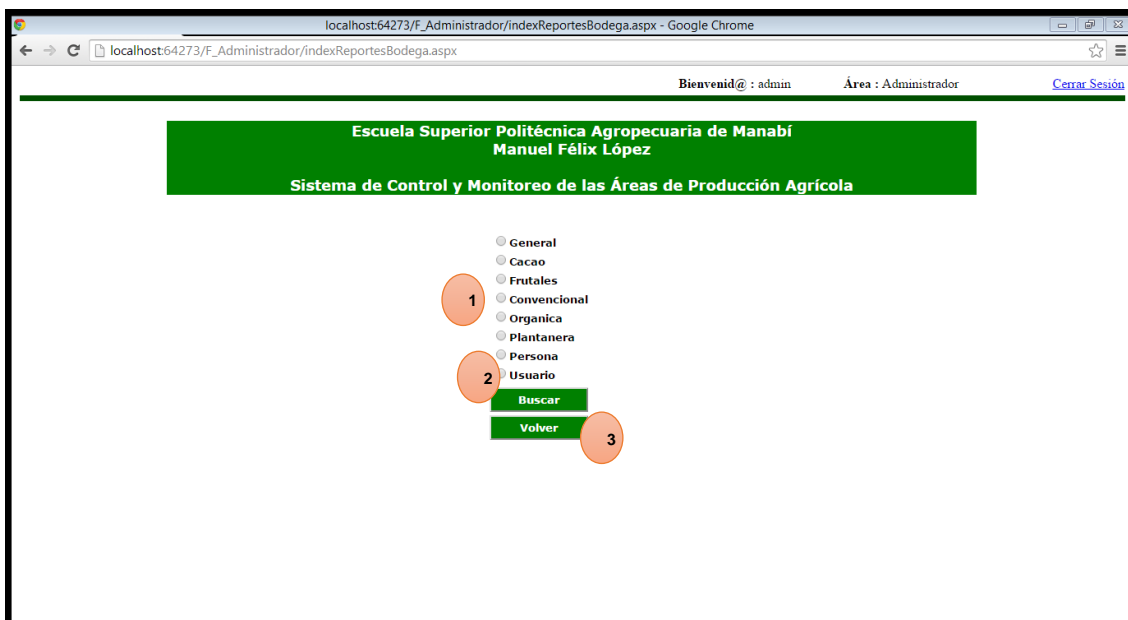
- 7) BOTON AGREGAR: Redirecciona al formulario de ingresar cultivo.
- 8) BOTÓN GUARDAR: Al dar clic almacena la siembra, cosecha, fertilización, postcosecha u otros según corresponda.

5. Uso para administrador



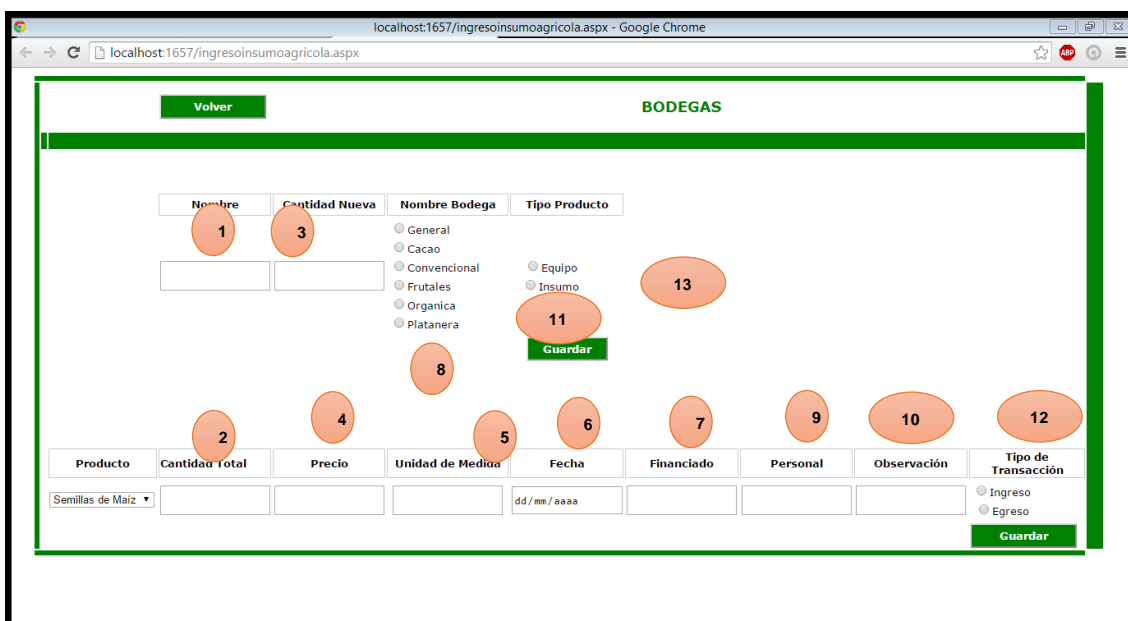
Ventana del index del administrador

- 1) MENÚ INICIO: Direcciona a la ventana inicio.
- 2) MENÚ METERELOGIA: Permite el ingreso y modificación de los datos metereologicos, calendario lunar y comunicados formato PDF.
- 3) INGRESOS: Permite el ingreso de personas, usuario y bodega.
- 4) MENÚ MODIFICAR: Permite modificar los datos de persona, usuario y bodega.
- 5) INFORMES: Muestra los informes de personas, usuario y bodega., además de las áreas de producción y cultivos.
- 6) SISTEMA: Muestra el mapa con acceso a todas las áreas.



Ventana de reportes de administrador

- 1) MENÚ DE SELECCIÓN DE INFORME: Selecciona el tipo de informe que desea cargar.
- 2) BOTÓN BUSCAR: Al dar clic carga el reporte.
- 3) BOTÓN VOLVER: Al dar clic regresa al index del Administrador.



Ventana de ingreso de bodega

- 1) Ingreso del nombre del producto o insumo a usar.

- 2) Ingreso de la cantidad a usar o a registrar en bodega.
- 3) Ingreso de la cantidad total de dicho producto en bodega.
- 4) Ingreso del precio del producto o insumo.
- 5) Ingreso de la unidad de medida.
- 6) Ingreso de la fecha del ingreso o egreso del producto o insumo.
- 7) Ingreso de quien financia el producto o insumo, ESPAM MFL o estudiantes.
- 8) Menú de selección del tipo de bodega.
- 9) Ingreso de la persona encargada del uso de dicho producto o insumo.
- 10) Ingreso de alguna observación.
- 11) Menú de selección si es producto o insumo.
- 12) Menú de selección del tipo de transacción, sea ingreso o egreso.
- 13) Botón de guardar: Al hacer clic se almacena el registro en bodega.

**Escuela Superior Politécnica Agropecuaria de Manabí
Manuel Félix López**

Sistema de Control y Monitoreo de las Áreas de Producción Agrícola

Bienvenid@ : admin Área : Administrador [Cerrar Sesión](#)

Cedula:

Nombres:

Apellidos:

Direccion:

Telefono:

Cargo:

[Guardar](#)
[Volver](#)

Id	CI	Nombres	Apellidos	Direccion	Telefono	Cargo
Modificar	1312020447	Gema Teresa	Mendoza Vera	Canuto	0991375705	Estudiante
Modificar	1312020504	Maria Paula	Zamora Mendoza	Calceta	0989898989	Director

Ventana de ingreso de persona

- 1) CEDULA: Ingreso de numero de cedula
- 2) NOMBRES: Ingreso de los nombres de la persona
- 3) APELLIDO: Ingreso de los apellidos de la persona
- 4) DIRECCIÓN: Ingreso de la dirección
- 5) TELÉFONO: Ingreso del número de teléfono

- 6) CARGO: Ingreso del cargo que tiene la persona en la institución
- 7) BOTÓN GUARDAR: Al dar clic se almacenan los datos ingresados
- 8) BOTÓN VOLVER: Al dar clic se direcciona a la ventana de index administrador
- 9) CONSULTA DE LAS PERSONAS: Muestra las personas ingresadas en el sistema, al dar clic en modificar direcciona a una nueva ventana para modificar los campos.

INGRESO DE USUARIOS

1 Nombre Usuario:

Contraseña: 2

3 Tipo Usuario:

Persona: 1312020447 4

5

6

	Id	Id Persona	Nombre Usuario	Contraseña	Tipo Usuario
7	Modificar	1	cacao	cacao	Cacao
	Modificar	1	frutales	frutales	Frutales
	Modificar	1	datos	datos	Meteorología
	Modificar	1	admin	admin	Administrador

Ventana de ingreso de usuario

- 1) NOMBRE DE USUARIO: Ingresar el nombre del usuario
- 2) CONTRASEÑA: Ingresar la contraseña para el nombre de usuario
- 3) TIPO DE USUARIO: Ingresar el tipo de usuario que solo puede ser Cacao, Frutales, Convencional, Orgánica, Platanera, Administrador de Campo, Director y Administrador.
- 4) PERSONA: Se selecciona el número de cedula de la persona que representa a ese usuario.
- 5) BOTÓN GUARDAR: Al dar clic se almacena el usuario.
- 6) BOTÓN VOLVER: Al dar clic direcciona a la ventana index administrador.
- 7) CONSULTA LOS USUARIOS INGRESADOS: Muestra los usuarios ingresados en el sistema, al dar clic en modificar direcciona a la ventana de modificación.

6. Datos meteorológicos

Bienvenid@ : datos Área : Meteorología

Inicio **Ingresos** **Modificar Datos Meteorológicos**

1 3 3

2



La agricultura es la madre fecunda que proporciona todas las materias primeras que dan movimiento a las artes y al comercio.
Manuel Belgrano

Misión

La Ingeniería Agrícola forma al profesional para que administre los procesos de la unidad de producción y su vinculación con el medio externo; dándole importancia al manejo racional de los recursos hídricos y del suelo, utilizando técnicas y principios acordes a la diversidad de agroecosistemas y aplicando métodos y tecnologías de producción agraria sostenible; e integrando al profesional a procesos globalizados de mercadeo y comercialización.

Visión

Convertirse en un ente dinamizador de las políticas agrarias del país atendiendo los retos de las tendencias globalizadoras de la sociedad actual con carácter inter y multidisciplinario para asegurar la explotación racional de los recursos hídricos, del suelo y la maquinaria agrícola que intervienen en el proceso agroproductivo.

Copyright © 2015 - Carrera de Agrícola - 052-686101 (Ext. 114)

Ventana de index meteorología

- 1) INICIO: direcciona a la página principal.
- 2) DATOS METEOROLÓGICOS: Direcciona a la ventana de ingreso de datos meteorológicos
- 3) CALENDARIO AGRÍCOLA Y COMUNICADOS: Direcciona a la ventana donde se cargan el calendario agrícola y lo boletines.

localhost:64273/F_Meteorologia/IngresoMeteorologico.aspx - Google Chrome
localhost:64273/F_Meteorologia/IngresoMeteorologico.aspx

Bienvenid@ : datos Área : Meteorologia Cerrar Sesión

**Escuela Superior Politécnica Agropecuaria de Manabí
Manuel Félix López**

Sistema de Control y Monitoreo de las Áreas de Producción Agrícola

Ingreso de Datos Meteorológicos

1 Fecha:

Viento: 2

3 Precipitación:

Temperatura Mínima: 4

5 Temperatura Máxima:

Humedad Relativa: 6

7 Heliofania:

Evaporación: 8

9 Nubosidad: 10

Guardar

Volver

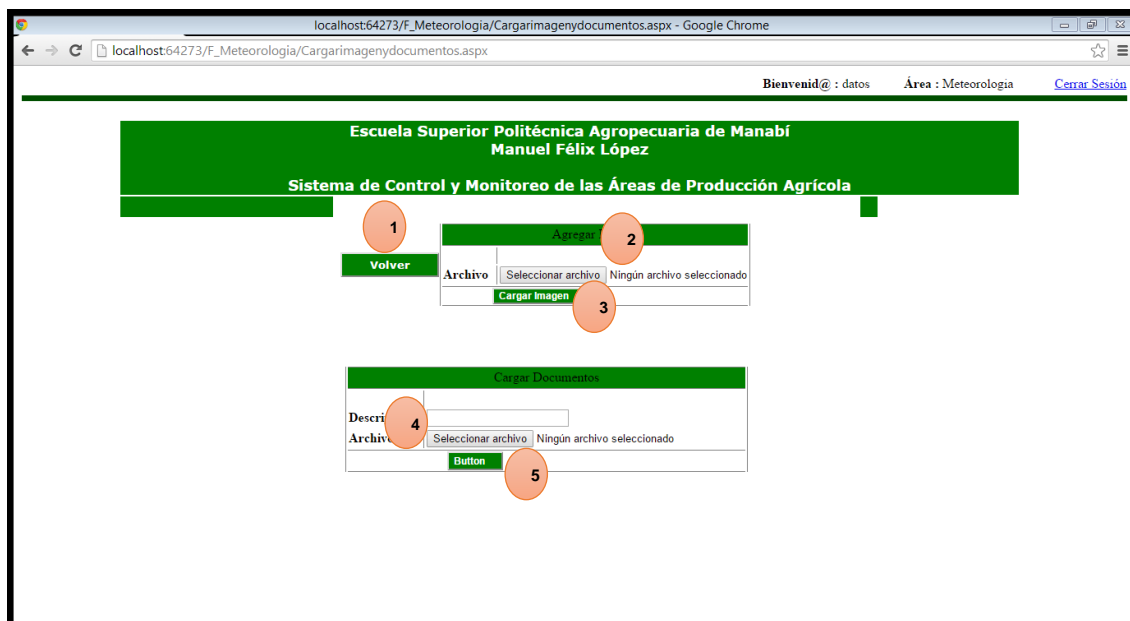
11

Ventana de ingreso de datos meteorológicos

- 1) FECHA: Ingresa la fecha
- 2) VIENTO: Ingresa el viento que se registró en el día
- 3) PRECIPITACIÓN: Ingresa la precipitación registrada en el día
- 4) TEMPERATURA MÁXIMA: Ingresa la temperatura máxima registrada en el día
- 5) TEMPERATURA MÍNIMA: Ingresa la temperatura mínima registrada en el día
- 6) HUMEDAD RELATIVA: Ingresa la humedad relativa registrada en el día
- 7) HELIOFANIA: Registra la heliofania registrada en el día
- 8) EVAPORACIÓN: Ingresa la evaporación registrada en el día
- 9) NUBOSIDAD: Registra la nubosidad registrada en el día
- 10) BOTÓN GUARDAR: Al dar clic se almacena la información registrada
- 11) BOTÓN VOLVER: Al dar clic direcciona a la ventana de index meteorología

NOTA: En los campos de precipitación, temperatura máxima y mínima, humedad relativa, heliofania, evaporación y nubosidad automáticamente se

registrara sin necesidad de ingresar sea mm o % en el caso de humedad relativa y así con los otros datos.




Ventana de agregar imagen y cargar documentos

- 1) BOTÓN VOLVER: Direcciona a la ventana index meteorología
- 2) SELECCIONAR ARCHIVO: Al dar clic se muestra una ventana donde se selecciona la imagen del calendario lunar, únicamente en formatos .jpg, .jpeg, .gif, .png, .JPG, .JPEG, .GIF y .PNG
- 3) BOTÓN CARGAR IMAGEN: Al dar clic se almacena la imagen cargada/
- 4) BOTÓN SELECCIONAR ARCHIVO: Al dar clic se muestra una ventana donde se selecciona un archivo únicamente en formato pdf.
- 5) BOTÓN CARGAR ARCHIVO: Al dar clic se almacena el archivo cargado.

7. Generar Informes

En el botón Informe de la Ventada de cada una de las áreas agrícolas dar clic y se redirecciona a la siguiente ventana:

Área : Cacao [Cerrar Sesión](#)
[Acerca de...](#)


 Inicio Calendario Datos Metereológicos Sistema Informes


Informes Cacao Clonal

Seleccionar la Labor Agrícola:

1 Siembra

Seleccionar las Fechas:

2 Desde: 

3 Hasta: 

Copyright © 2015 - Carrera de Agrícola - 052-686101 (Ext. 114)

Pantalla de vista del informe

- 1) Selección de labor agronómica
- 2) Selección del rango de fechas
- 3) Selección del rango de fechas

8. Contactos

Correo electrónico: agricola@espam.edu.ec

Teléfono: 052686101 ext. 114