



Universidad de Cienfuegos
Facultad de Ingeniería
Carrera de Ingeniería Informática

Título:

Sistema informático para el control de la información de las plagas y enfermedades de los cultivos en la provincia de Cienfuegos.

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:

Basilio Yoel Gómez Guerra

Tutor:

Ms. C. Kadir Héctor Ortiz

Cienfuegos, Cuba

Curso 2017 – 2018

Declaración de autoría

Yo Basilio Yoel Gómez Guerra declaro que soy el único autor de este trabajo de diploma titulado: Sistema Informático para el control de la información de plagas y enfermedades de los cultivos en la provincia de Cienfuegos y autorizo a la Facultad de Ingeniería de la Universidad de Cienfuegos “Carlos Rafael Rodríguez” y al Centro de Estudios de Ingeniería y Sistemas para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del _____.

Firma Autor

Basilio Yoel Gómez Guerra

Firma Tutor

Ms. C. Kadir Héctor Ortiz

Dedicatoria

A mi familia.

Agradecimientos

- *A mis padres que me dieron siempre su apoyo incondicional sin importar los malos momentos to' el tiempo pa'arriba del lio.*
- *A mi hermano Jansel el Cabezón que me tiró el cabo cuando más lo necesitaba, sin pedir nada a cambio.*
- *A Sandy (que sin susto es) el dios de la programación al que tampoco tendré como agradecerle nunca.*
- *A mi novia Yesenia, que me dio amor y cariño en los momentos de mayor estrés, y a su familia que al igual que la mía, me acogieron como uno más de los suyos e hicieron mucho por mí sin escatimar.*
- *A mis socios de cuarto y los todos los que conocí durante todo este tiempo de situaciones adversas y de dicha, que siempre se presentaron cuando los necesité y cuando no también.*
- *A mis amigos del plantel de La FEU.*
- *A Kadir mi tutor que me regañó y me halagó cuando lo tuvo que hacer, además de hacerse cargo de mí cuando nadie más lo hizo, y que mucho me ayudó de manera indescriptible.*

- *A todos los que de alguna forma u otra hicieron posible mi formación tanto espiritual como profesionalmente.*

Resumen

El presente trabajo tiene por título “Sistema informático para la gestión de la información referente al comportamiento de las plagas y enfermedades de los cultivos en la provincia de Cienfuegos”, se desarrolló en el departamento de Sanidad Vegetal perteneciente al Ministerio de la Agricultura de Cienfuegos (MINAG) y tuvo como fin la implementación de un sistema informático para el control de la información sobre el comportamiento de las plagas y enfermedades de los cultivos en la provincia de Cienfuegos, con el fin de mejorar el proceso de gestión de enfermedades y plagas que puedan presentar los diferentes tipos de cultivos. Ofrece además un conjunto de reportes que contribuyen a la toma de decisiones en la empresa.

Se empleó para el desarrollo de la aplicación PHP como lenguaje de programación, VisualParadigm_Suite_Windows_5_0 para Windows además de MySQL Server como Sistema Gestor de Bases de Datos, además de los frameworks Bootstrap para las vistas y Symfony como Framework de programación.

Palabras claves: Sistema de informático, control de la información, proceso de gestión.

Abstract

The present work is entitled "Computer system for the management of information related to the behavior of diseases and diseases of crops in the province of Cienfuegos", in the Department of Plant Health belonging to the Ministry of Agriculture of Cienfuegos (MINAG) and aimed at the implementation of a computer system for the control of information on the behavior of diseases and diseases of crops in the province of Cienfuegos, in order to improve the process of management of diseases and pests that may occur the different types of crops. It also offers a set of reports that contribute to the decision making in the company.

It was used to develop the PHP application as a programming language, VisualParadigm_Suite_Windows_5_0 for Windows as well as MySQL Server as a Database Management System, as well as Bootstrap frameworks for views and Symfony as a programming framework.

Keywords: Computer system, information control, management process.

Índice

<i>Introducción</i>	1
<i>Capítulo I: Fundamentación teórica</i>	5
1.1 Uso de las Tecnologías de la Información y las Comunicaciones (TIC) en los servicios de la Agricultura.	5
1.2 Conceptos asociados al dominio del sistema.	5
1.3 Flujo del proceso y análisis crítico.	6
1.4 Análisis de los sistemas informáticos existentes.	7
1.4.1 Sistemas informáticos existentes a nivel internacional.	7
1.4.2 Sistemas informáticos existentes nacionalmente.	7
1.4.3 Análisis comparativo de las soluciones existentes con la propuesta.	7
1.5 Metodologías de Desarrollo de Software.	8
1.5.1 El Proceso Unificado de Desarrollo de Software (RUP).	10
1.6. Tecnologías y lenguajes.	11
1.6.1 Arquitectura del sistema.	11
1.6.2 Arquitectura de N capas.	12
1.7 Patrón de arquitectura de software.	13
1.7.1 Patrón de diseño MVC.	14
1.8 Sistema Gestor de Base de Datos.	14
1.8.1 MySQL:	15
Es un sistema de administración de bases de datos (<i>Database Management System, DBMS</i>) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.....	15
1.9 Lenguajes utilizados en el desarrollo del sistema.	15

Índice

1.10 Servidores de aplicaciones.....	17
1.11 Framework, librerías y componentes.	18
1.12 Herramientas utilizadas.	21
1.13 Conclusiones del capítulo.	21
<i>Capítulo II: Construcción del Sistema Informático Propuesto.....</i>	<i>27</i>
Introducción al capítulo.	27
2.1 Breve descripción de los procesos del negocio.....	27
2.2 Reglas del negocio.....	27
2.3 Modelo de casos de uso del negocio.	28
2.3.1 Actores del negocio.....	28
2.3.2 Trabajadores del negocio.....	28
2.3.3 Diagrama de casos de uso del negocio.	29
2.3.4 Descripciones de los casos de uso del negocio.....	29
2.4 Modelo de objeto del negocio.	30
2.5 Descripción del modelo del sistema.....	30
2.5.1 Requerimientos funcionales.	30
2.5.2 Requerimientos no funcionales.....	32
2.6 Modelo de Caso de Uso del Sistema.	34
2.6.1 Actores del Sistema.....	34
2.6.2 Casos de uso del sistema.	34
2.6.3 Diagrama de Casos de Uso del Sistema.	35
2.6.4 Descripción de los casos de uso del sistema	36
2.7 Construcción del Sistema.....	36
2.7.1 Diagramas de clases del diseño.....	36

Índice

2.8 Diseño de la Base de Datos.	37
2.8.1 Modelo lógico de datos.	37
2.8.2 Modelo físico de datos.	38
2.9 Modelo de despliegue.	39
2.9 Principios de diseño.	40
2.9.1 Interfaz de usuario.	40
2.9.2 Tratamiento de excepciones.	41
2.10 Conclusiones del capítulo.	41
<i>Capítulo III: Estudio de factibilidad y casos de pruebas funcionales.</i>	44
Introducción al capítulo.	44
3.1 Estudio de factibilidad.	44
3.1.1 Puntos de casos de uso sin ajustar.	44
3.1.2 Puntos de Casos de Uso Ajustados.	47
3.1.3 Estimación del esfuerzo.	49
3.1.4 Cálculo de costo.	51
3.2 Análisis de los beneficios tangibles e intangibles.	51
3.3 Casos de Pruebas Funcionales.	51
3.3.1 Caso de Prueba para Autenticarse.	52
3.3.2 Caso de Prueba para Cambiar contraseña.	53
3.3.3 Caso de Prueba para Gestionar Usuario.	54
3.3.4 Caso de Prueba para Gestionar Modelo.	55
3.3.5 Caso de Prueba para Gestionar Empresa.	56
3.3.6 Caso de Prueba para Gestionar Rama.	57
3.3.7 Caso de Prueba para Gestionar EPP.	58

Índice

3.3.8 Caso de Prueba para Gestionar Unidad.....	59
3.3.9 Caso de Prueba para Gestionar Municipio.....	60
3.3.10 Caso de Prueba para Gestionar Cultivo.....	61
3.3.11 Caso de Prueba para Gestionar Agente Nocivo.....	62
3.3.12 Caso de Prueba para Gestionar Producto.....	63
Conclusiones del capítulo.....	64
<i>Conclusiones.....</i>	<i>65</i>
<i>Recomendaciones.....</i>	<i>65</i>
<i>Referencias bibliográficas.....</i>	<i>67</i>
<i>Bibliografía.....</i>	<i>1</i>
<i>Glosario de términos.....</i>	<i>1</i>
<i>Anexos.....</i>	<i>1</i>
Anexo A: Descripción de los Casos de Uso del Sistema.....	1
A.1 Descripción textual del caso de uso Autenticarse.....	1
A.2 Descripción textual del Caso de Cerrar Sesión.....	1
A.3 Descripción textual del Caso de Uso Cambiar Contraseña.....	1
A.4 Descripción textual del Caso de Uso Gestionar Usuario.....	2
A.5 Descripción textual del Caso de Uso Gestionar Municipio.....	2
A.6 Descripción textual del Caso de Uso Gestionar Empresa.....	2
A.7 Descripción textual del Caso de Uso Gestionar Rama.....	3
A.8 Descripción textual del Caso de Uso Gestionar Unidad.....	3
A.9 Descripción textual del Caso de Uso Gestionar Estación.....	3
A.10 Descripción textual del Caso de Uso Gestionar Cultivo.....	4
A.11 Descripción textual del Caso de Uso Gestionar Agente Nocivo.....	4

Índice

A.12 Descripción textual del Caso de Uso Gestionar Modelo.....	5
A.13 Descripción textual del Caso de Uso Visualizar Modelo 2004.....	5
A.14 Descripción textual del Caso de Uso Visualizar Modelo 2004d.	5
A.15 Descripción textual del Caso de Uso Visualizar Modelo 2003.....	6
Anexo B: Prototipos de Interfaz.	6
Anexo P1: Autenticarse.....	6
Anexo P2: Cambiar Contraseña.....	7
Anexo P3: Gestionar Usuario.....	7
Anexo P4: Gestionar Municipio.	8
Anexo P5: Gestionar Empresa.....	8
Anexo P6: Gestionar Rama.	8
Anexo P7: Gestionar Unidad.....	9
Anexo P8: Gestionar Epp.	9
Anexo P9: Gestionar Cultivo.....	10
Anexo P10: Gestionar Agente Nocivo.....	10
Anexo P11: Gestionar Producto.	10
Anexo P12: Gestionar Modelo 2004.	11
Anexo P13: Gestionar Modelo 2004d.	11
Anexo P14: Gestionar Modelo 2003.	11
Anexo C: Diagramas de Clase Web.	12
Anexo C1: Autenticarse.....	12
Anexo C2: Cerrar sesión.	12
Anexo C3: Cambiar contraseña.....	13
Anexo C4: Gestionar Usuario.	13

Índice

Anexo C5: Gestionar Municipio.....	14
Anexo C6: Gestionar Empresa.....	14
Anexo C7: Gestionar Rama.....	15
Anexo C8: Gestionar Unidad.....	15
Anexo C9: Gestionar EPP.	16
Anexo C10: Gestionar Cultivo.....	16
Anexo C11: Agente Nocivo.	17
Anexo C12: Gestionar Modelo 2004.	17
Anexo C13: Gestionar Modelo 2004d.	18
Anexo C14: Gestionar Modelo 2003.	18

Introducción

El Ministerio de la Agricultura es el organismo de la Administración Central del Estado que tiene la misión de proponer y una vez aprobada dirigir, ejecutar y controlar la política del Estado y Gobierno sobre el uso, tenencia y explotación sostenible y sustentable de la superficie agrícola del país propiedad de todo el pueblo, colectiva e individual, la producción agropecuaria y forestal para la satisfacción de las necesidades alimentarias de la población, la industria y la exportación. El sistema organizativo atendido por el MINAG está integrado por una esfera estatal y otra empresarial y cooperativa. La esfera estatal presupuestada, está compuesta por el Órgano Central del Ministerio, las entidades adscriptas y las delegaciones provinciales de la Agricultura. La esfera empresarial y cooperativa, está integrada por organizaciones superiores de dirección empresarial, empresas, granjas estatales y unidades productoras, integradas por UBPC, CPA y CCS.

Entre sus principales órganos de dirección se encuentra la Dirección General de Agricultura, con las Direcciones de Agricultura; Sanidad Vegetal; Forestal, Flora y Fauna Silvestre; Semillas y Recursos Fitogenéticos. Dirección General de Ganadería, con las Direcciones de Ganadería; Genética y Registro Pecuario; Sanidad Animal.

El Departamento Provincial de Sanidad Vegetal está compuesto por: Dirección Provincial Sección de: Protección de Plantas, Cuarentena Vegetal, Desarrollo y Servicios Técnicos para la Sanidad Vegetal, Laboratorio Provincial de Sanidad Vegetal, Estaciones de Protección de Plantas (EPP) y Punto Frontera). El sistema estatal de protección de plantas en Cuba fue creado en 1972 y rige la política, decretos y estrategias para el control de plagas en el país y consta de: Una Dirección General, 16 Departamentos Provinciales, 65 Estaciones Territoriales de Protección de Plantas, 14 Laboratorios Provinciales y 24 Puntos Fronteras. Las Estaciones Territoriales de Protección de Plantas constan de: Grupo de Señalización y Pronóstico, Desarrollo y Servicios Técnicos para la Sanidad Vegetal, Grupo de Cuarentena Vegetal y Protección de Plantas.

Las funciones de las Estaciones de Protección de Plantas (EPP) son:

- El monitoreo y las acciones de control de las plagas en los cultivos.

Introducción

- Elaborar y emitir oportunamente la señal sobre la aparición y desarrollo de organismos nocivos en su territorio teniendo en cuenta los conocimientos históricos, la información obtenida en los campos estacionarios y los recorridos itinerarios.
- Elaborar los registros territoriales históricos a partir de la información básica obtenida.

Teniendo en cuenta lo antes planteado se expone la siguiente **situación problemática**:

Debido a que actualmente existen en la provincia varias estaciones de protección y la gran cantidad de plagas de cultivos, el control y el estudio de la información de las mismas se complejiza ya que este proceso en la actualidad se realiza en parte de forma manual y en parte es almacenada por medio de hojas de cálculo Excel lo cual ralentiza el trabajo y propicia la pérdida de información, falta de integridad e inconsistencia de la misma.

Teniendo en cuenta estas afirmaciones, se define como **problema a resolver**:

¿Cómo mejorar el proceso de gestión de la información de las plagas y enfermedades en los cultivos en la provincia de Cienfuegos?

Identificando como **objeto de estudio**, La gestión de la información en el departamento de Sanidad Vegetal en la provincia de Cienfuegos y como **campo de acción** la información sobre las plagas y enfermedades de los cultivos en el departamento de Sanidad Vegetal en la provincia de Cienfuegos.

Resulta de vital importancia la gestión, organización y control del flujo de la información en la entidad, a la hora de la toma de cualquier decisión recayendo así en dicha institución una gran responsabilidad teniendo en cuenta los puntos de integridad, fiabilidad, disponibilidad y seguridad. Por tanto, es necesario que exista un correcto intercambio de información y comunicación que haga el trabajo más sencillo y eficaz, por lo que se tiene una posible solución a los problemas existentes a las tecnologías de la información y las comunicaciones (TIC). El uso de las TIC es una opción necesaria en el desarrollo de la sociedad, significando un cambio notable cuando son puestas en práctica en los diferentes sectores e instituciones, pues sin lugar a dudas las mejoras son sustanciales en cuanto a tiempo y recursos se refiere.

Introducción

Como **idea a defenderse** define que el desarrollo de un sistema informático en el Departamento de Sanidad Vegetal mejorará la gestión de la información del proceso fitosanitario.

Para solucionar el problema planteado anteriormente se define como **objetivo general**:

Desarrollar un sistema informático que procese la información sobre las plagas y enfermedades de los cultivos en la provincia de Cienfuegos.

A raíz del objetivo general se trazaron los siguientes **objetivos específicos**:

- Elaborar un marco teórico conceptual que sirva de referencia para el desarrollo metodológico del trabajo.
- Modelar el negocio, describir el flujo de la información de los procesos y diseñar un sistema informático para una mejor comprensión de estos.
- Implementar una aplicación para gestionar la información referente al comportamiento de las plagas y enfermedades de los cultivos.
- Validar el sistema.

Definiendo como **aporte práctico** el sistema informático que facilita la obtención de información sobre las plagas y enfermedades de los cultivos en el Departamento de Sanidad Vegetal en la provincia de Cienfuegos.

El presente trabajo tiene una estructura que consta de tres capítulos detallados a continuación:

Capítulo I: Fundamentación Teórica:

Donde se explica la fundamentación teórica del tema y los conceptos asociados al problema en cuestión, el problema a resolver y el campo de acción en que se desarrolla, las tecnologías, lenguajes y metodologías que se utilizarán para su desarrollo.

Capítulo II: Descripción y construcción de la solución propuesta:

Donde se explica el proceso de construcción para solucionar la problemática planteada. Describe a groso modo cómo funciona la aplicación. Define requisitos funcionales y no funcionales y detalla el modelo de casos de uso, el diseño y la implementación del sistema.

Capítulo III: Estudio de factibilidad y diseño de casos de prueba funcional:

Donde se realiza un estudio de factibilidad del sistema propuesto, en cuanto a costos, beneficios y planificación del mismo se refiere. Se diseñan además los casos de prueba funcional del sistema y se especifica la manera correcta de su funcionamiento.

Capítulo I: Fundamentación teórica

Introducción del Capítulo.

En este capítulo se propone la base teórica y conceptual para desarrollar el sistema de gestión de la información referente al comportamiento de las plagas y enfermedades en los cultivos en la provincia de Cienfuegos. Como temas más importantes se estudian los principales conceptos que se relacionan con el campo de acción y el dominio del problema, los antecedentes de sistemas relacionados con el tema, existentes tanto internacional como nacionalmente. Además, se estudian las metodologías, tecnologías, herramientas y lenguajes y se seleccionan los más adecuados para el desarrollo del mismo.

1.1 Uso de las Tecnologías de la Información y las Comunicaciones (TIC) en los servicios de la Agricultura.

Conocidas con las siglas TIC, son el conjunto de medios (radio, televisión y telefonía convencional) de comunicación y las aplicaciones de información que permiten la captura, producción, almacenamiento, tratamiento, y presentación de informaciones en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética. Las TIC incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual.[1] Dentro de las tecnologías de la información y la comunicación (TIC), se tienen los elementos y las técnicas utilizadas en el tratamiento y transmisión de la información, los cuales agilizan gradualmente el procesamiento de la información y la comunicación.

1.2 Conceptos asociados al dominio del sistema.

El Ministerio de la Agricultura es el organismo de la Administración central del Estado que tiene la misión de proponer y una vez aprobada dirigir, ejecutar y controlar la política del Estado y Gobierno sobre el uso, tenencia y explotación sostenible y sustentable de la superficie agrícola del país, propiedad de todo el pueblo, colectiva e individual; la producción agropecuaria y forestal para la satisfacción de las necesidades alimentarias de la población, la industria y la exportación.

Plaga: Se conoce como plaga a la irrupción súbita y multitudinaria de insectos, animales u otros organismos de una misma especie que provoca diversos tipos de perjuicios. Cuando

Capítulo I: Fundamentación teórica

se registran a gran escala y afectan cultivos, este tipo de plagas pueden destruir cosechas enteras e impedir el normal desarrollo de la agricultura.[2]

Cultivo: Es una serie de técnicas que se aplican para lograr los mayores frutos de la tierra o de la capacidad humana.[3]

Vigilancia Fitosanitaria: Es la observación sistemática de un producto o área para detectar presencia o ausencia de plagas o el comportamiento de éstas, dentro de un espacio geográfico y un tiempo determinados.[4]

1.3 Flujo del proceso y análisis crítico.

El principal objetivo del departamento de Sanidad Vegetal es la Vigilancia Fitosanitaria, señalizando y pronosticando el comportamiento de las plagas y enfermedades en los diversos tipos de cultivos, captando así y procesando la información manejada por las EPP logrando el correcto desarrollo y crecimiento de dichos cultivos.

El Director Provincial solicita al final de cada mes el informe de control de plagas al consultor, acto seguido este le solicita la información necesaria al analista. En las EPP ubicadas en los diferentes municipios de la provincia el analista recopila a través de estudios sistemáticos la información concerniente al crecimiento de los cultivos en los campos y las plagas o enfermedades que puedan o no presentarse, luego este envía la información recogida a lo largo del período mensual, al consultor en el Departamento Provincial de sanidad Vegetal del MINAG, este proceso es realizado en parte de forma manual, y a través de una hoja de cálculo Excel.

Una vez recibida la información, el consultor primeramente chequea que dicha información esté totalmente correcta, luego clasifica, si esta presenta algún error la información es devuelta al analista para que este la corrija, de estar correcta procede a realizar los diferentes modelos de análisis atendiendo a la índole de los datos recibidos. Una vez realizados estos modelos, son estudiados minuciosamente para poder dar un diagnóstico exacto y proponer una posible solución a los problemas actuales la cual es entregada al director provincial en conjunto con un informe con todos los datos para que este disponga de ellos, y se encargue de aprobar o no las decisiones tomadas.

Capítulo I: Fundamentación teórica

1.4 Análisis de los sistemas informáticos existentes.

En el mundo existen varios softwares relacionados con la gestión de la sanidad vegetal dando así una muestra de la importancia de las TIC en el campo de la agricultura. A continuación, se muestran algunos de ellos:

1.4.1 Sistemas informáticos existentes a nivel internacional.

SAVIA: Es una aplicación web de soporte a la decisión en el control de plagas y enfermedades en el cultivo de la uva de mesa. Para su construcción se ha aplicado un método de desarrollo de software basado en la metodología de desarrollo de software dirigido por modelos, que permite la generación automática de la aplicación web a partir del modelo conceptual del problema.[5]

AgroNews: Es una aplicación Androide, que se encarga de brindar posibles soluciones y pronósticos a partir de la información recopilada por dispositivos satelitales con bases de conocimientos propias, sobre la posible presencia de plagas y enfermedades en los cultivos. Esta aplicación es de carácter gratuito y se adecua perfectamente a las especificaciones de la empresa, pero los implementos utilizados presentan un alto costo adquisitivo.[6]

1.4.2 Sistemas informáticos existentes nacionalmente.

A nivel nacional, se considera que el sector de la agricultura se encuentra en una fase de inicio aún, ya que no todas las ramas de este sector se encuentran fusionadas con la tecnología, y la falta de softwares de captación y procesamiento de la información que es manejada por las EPP y los Departamentos Provinciales de Sanidad Vegetal es una muestra clara de ello.

1.4.3 Análisis comparativo de las soluciones existentes con la propuesta.

Después de haber realizado un estudio detallado de los sistemas existentes, se concluye que ninguna de las soluciones encontradas son candidatas a ser utilizadas pues ninguna de ellas se adapta concretamente a las necesidades específicas del proceso de gestión de la información manejada. Teniendo en cuenta tales necesidades se decidió realizar un sistema informático que gestione la información referente al comportamiento de las plagas y enfermedades de los cultivos. La solución propuesta tendrá como principal ventaja que se adaptará perfectamente a características del sector, cumpliendo con todas las necesidades

Capítulo I: Fundamentación teórica

asociadas a gestión de la información referente al comportamiento de las plagas y enfermedades de los cultivos.

1.5 Metodologías de Desarrollo de Software.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados.[7]

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo. Las metodologías se basan en una combinación de los modelos de proceso genéricos, cascada, incremental, etc. Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas.[7]

Son muchas las ventajas que puede aportar el uso de una metodología, A continuación, se van a exponer algunas de ellas, clasificadas desde distintos puntos de vista.

- Desde el punto de vista de gestión:
- Facilitar la tarea de planificación.
- Facilitar la tarea del control y seguimiento de un proyecto.
- Mejorar la relación coste/beneficio.
- Optimizar el uso de recursos disponibles.
- Facilitar la evaluación de resultados y cumplimiento de los objetivos.
- Facilitar la comunicación efectiva entre usuarios y desarrolladores.

Desde el punto de vista de los ingenieros del software:

- Ayudar a la comprensión del problema.

Capítulo I: Fundamentación teórica

- Optimizar el conjunto y cada una de las fases del proceso de desarrollo.
- Facilitar el mantenimiento del producto final.
- Permitir la reutilización de partes del producto.

Desde el punto de vista del cliente o usuario:

- Garantía de un determinado nivel de calidad en el producto final.
- Confianza en los plazos de tiempo fijados en la definición del proyecto.

Definir el ciclo de vida que más se adecue a las condiciones y características del desarrollo. En general las metodologías llevan a cabo una serie de procesos comunes que son buenas prácticas para lograr los objetivos antes mencionados independientemente de cómo hayan sido diseñadas. Las fases que agrupan estos procesos son las siguientes:[7]

- Análisis
- Especificación
- Diseño
- Programación
- Prueba
- Documentación
- Mantenimiento
- Reingeniería

Metodologías tradicionales

Las metodologías tradicionales son denominadas, a veces, de forma peyorativa, como metodologías pesadas. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil. Las metodologías tradicionales (formales) se focalizan en la documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, etc.).[7]

Metodologías ágiles

Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan.[7]

Debido a esto la selección de una metodología para el desarrollo de un software se hace necesario un estudio de las características del software y del equipo.

Asumiendo, que el software después de terminado, podrá ser sometido a nuevas reestructuraciones y cambios, se necesita que la documentación sea muy detallada, para facilitar el trabajo futuro, por lo que se opta por utilizar la metodología de desarrollo de software Rational Unified Process (RUP).

1.5.1 El Proceso Unificado de Desarrollo de Software (RUP).

Rational Unified Process o Proceso Unificado de Racional, es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo e incremental que es enfocada hacia diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura como tal. Utiliza el Lenguaje Unificado de Modelado (UML).[8]

Tiene como **características**:[8]

- ❖ Está dirigido por casos de uso.
- ❖ Está centrado en la arquitectura.
- ❖ Es iterativo e incremental.
- ❖ Es amplio y diverso.
- ❖ Su enfoque es orientado a objetos.

Como principales ventajas se tienen:

- Es el proceso de desarrollo más general de los existentes actualmente.

Capítulo I: Fundamentación teórica

- Promueve la reusabilidad.
- Reduce la complejidad del mantenimiento (extensibilidad y facilidad de cambios).
- Riqueza semántica.
- Disminuye la brecha semántica entre la visión interna y la visión externa del sistema.
- Facilita la construcción de prototipos.

1.6. Tecnologías y lenguajes.

A continuación, se exponen las características fundamentales de las tecnologías y lenguajes a utilizar en el desarrollo del sistema.

1.6.1 Arquitectura del sistema.

Arquitectura cliente-servidor.

La arquitectura cliente-servidor en la actualidad es una de las más importantes y utilizadas en el ámbito de enviar y recibir información, también es una herramienta potente para guardar los datos en una base de datos como servidor. Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores.[9]

Como principales **características** se tienen:[9]

- ❖ El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- ❖ Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- ❖ Un servidor da servicio a múltiples clientes en forma concurrente.
- ❖ Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

Capítulo I: Fundamentación teórica

- ❖ La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- ❖ Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.
- ❖ También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.
- ❖ Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.
- ❖ Además, se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.
- ❖ Designa un modelo de construcción de sistemas informáticos de carácter distribuido.
- ❖ Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

1.6.2 Arquitectura de N capas.

Este estilo de despliegue arquitectónico describe la separación de la funcionalidad en segmentos separados de forma muy parecida al estilo de capas, pero en el cual cada segmento está localizado en un computador físicamente separado. Este estilo ha evolucionado desde la aproximación basada en componentes generalmente usando métodos específicos de comunicación asociados a una plataforma en vez de la aproximación basada en mensajes.[11]

Como principales ventajas del estilo de arquitectura de N-capas/3-capas son:[11]

Capítulo I: Fundamentación teórica

Mejoras en las posibilidades de mantenimiento. Debido a que cada capa es independiente de la otra los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.

Escalabilidad: Como las capas están basadas en diferentes maquinas, el escalamiento de la aplicación hacia afuera es razonablemente sencillo.

Flexibilidad: Como cada capa puede ser manejada y escalada de forma independiente, la flexibilidad se incrementa.

Disponibilidad: Las aplicaciones pueden aprovechar la arquitectura modular de los sistemas habilitados usando componentes que escalan fácilmente lo que incrementa la disponibilidad.

1.7 Patrón de arquitectura de software.

Los patrones de arquitectura están orientados a representar los diferentes elementos que componen una solución de software y las relaciones entre ellos. A diferencia de los patrones de diseño de software que están orientados a objetos y clases (patrones creacionales, estructurales, de comportamiento, de interacción, etc.), los patrones de arquitectura están a un mayor nivel de abstracción. Los patrones de arquitectura forman parte de la llamada Arquitectura de Software (arquitectura lógica de un sistema), que de forma resumida comprende:

- El diseño de más alto nivel de la estructura del sistema.
- Los patrones y abstracciones necesarios para guiar la construcción del software de un sistema.
- Los fundamentos para que analistas, diseñadores, programadores, beta testers, etc. trabajen en una línea común que permita cubrir restricciones y alcanzar los objetivos del sistema.
- Los objetivos del sistema, no solamente funcionales, sino de mantenimiento, auditoría, flexibilidad e interacción con otros sistemas.
- Las restricciones que limitan la construcción del sistema acorde a las tecnologías disponibles para su implementación.

1.7.1 Patrón de diseño MVC.

El patrón Modelo-Vista-Controlador (MVC) define la organización independiente del **Modelo** (Objetos de Negocio), la **Vista** (interfaz con el usuario u otro sistema) y el **Controlador** (controlador del workflow de la aplicación). De esta forma, se divide el sistema en tres capas donde, se tiene la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador.[12]

El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:[12]

Modelo

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- No sabe nada / independiente del Controlador y la Vista.

Vista

- Es la presentación del Modelo.
- Puede acceder al Modelo, pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo.

Controlador

- Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente.

1.8 Sistema Gestor de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el

Capítulo I: Fundamentación teórica

objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.[13]

1.8.1 MySQL:

Es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.[14]

1.9 Lenguajes utilizados en el desarrollo del sistema.

UML: Es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un *método* es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Un modelo es expresado en un *lenguaje de modelado*. Un lenguaje de modelado consiste de vistas, diagramas, elementos de modelo $\frac{3}{4}$ los símbolos utilizados en los modelos $\frac{3}{4}$ y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas.[15]

Capítulo I: Fundamentación teórica

HTML: Es el lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc. Además, sirve para crear páginas web, darles estructura y contenido.[16]

CSS: Es un lenguaje utilizado en la presentación de documentos HTML. Este sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc. La filosofía de CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación. O sea, la página web sería lo que hay debajo (el contenido) y CSS sería un cristal de color que hace que el contenido se vea de una forma u otra.[17]

PHP: (siglas que originalmente significaban Personal Home Page) fue primero escrito por Rasmus Lerdorf como un simple conjunto de scripts de Perl para guiar a los usuarios en sus páginas. Luego para satisfacer inquietudes del mismo tipo por parte de otra gente lo reescribe, pero esta vez como un lenguaje de script agregándole entre otras características soporte para formularios. Al ver como la popularidad del lenguaje aumenta, un grupo de desarrolladores crea para él un API, convirtiéndose así en el PHP3. Fue en ese momento cuando el parser de scripts PHP es completamente reescrito (el Zend Engine) dando vida al PHP4 mucho más rápido y luego al PHP5.[18]

Las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad.[18]

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs

Capítulo I: Fundamentación teórica

sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo `.ini`.
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.
- **Conectividad:** PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

En el desarrollo de la herramienta se utilizó PHP5 en correspondencia con la versión de PHP soportada por el framework utilizado para la construcción de la propuesta.

1.10 Servidores de aplicaciones.

El concepto de servidor de aplicaciones está relacionado con el concepto de sistema distribuido. Un sistema distribuido, en oposición a un sistema monolítico, permite mejorar tres aspectos fundamentales en una aplicación: la alta disponibilidad, la escalabilidad y el mantenimiento. En un sistema monolítico un cambio en las necesidades del sistema (aumento considerable del número de visitas, aumento del número de aplicaciones, etc.) provoca un colapso y la adaptación a dicho cambio puede resultar catastrófica.[19]

La alta disponibilidad hace referencia a que un sistema debe estar funcionando las 24 horas del día los 365 días al año. Para poder alcanzar esta característica es necesario el uso de técnicas de balanceo de carga y de recuperación ante fallos (failover).

La escalabilidad es la capacidad de hacer crecer un sistema cuando se incrementa la carga de trabajo (el número de peticiones). Cada máquina tiene una capacidad finita de recursos y por lo tanto sólo puede servir un número limitado de peticiones. Si, por ejemplo, tenemos una tienda que incrementa la demanda de servicio, debemos ser capaces de incorporar nuevas máquinas para dar servicio.

Capítulo I: Fundamentación teórica

El mantenimiento tiene que ver con la versatilidad a la hora de actualizar, depurar fallos y mantener un sistema. La solución al mantenimiento es la construcción de la lógica de negocio en unidades reusables y modulares.

Un servidor de aplicaciones proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas. Es el corazón de un gran sistema distribuido.

Servidor web Apache:El servidor Apache HTTP, también llamado **Apache**, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.[20]

1.11 Framework, librerías y componentes.

Es un marco de aplicación o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones. En esta categoría se incluye la Biblioteca de Componentes Visuales (VCL) de CodeGear, Swing y AWT para el desarrollo de aplicaciones con formularios en Java, Struts, para aplicaciones web también en Java, Ruby On Rails para aplicaciones web con Ruby, y las antiguas Microsoft Foundation Classes y Turbo Vision de Borland.[21]

Las principales ventajas de la utilización de un framework son:

- El desarrollo rápido de aplicaciones. Los componentes incluidos en un framework constituyen una capa que libera al programador de la escritura de código de bajo nivel.
- La reutilización de componentes software al por mayor. Los frameworks son los paradigmas de la reutilización.
- El uso y la programación de componentes que siguen una política de diseño uniforme. Un framework orientado a objetos logra que los componentes sean clases que pertenezcan a una gran jerarquía de clases, lo que resulta en bibliotecas más fáciles de aprender a usar.

Las desventajas de los frameworks son:

- La dependencia del código fuente de una aplicación con respecto al framework. Si se desea cambiar de framework, la mayor parte del código debe reescribirse.

Capítulo I: Fundamentación teórica

- La demanda de grandes cantidades de recursos computacionales debido a que la característica de reutilización de los frameworks tiende a generalizar la funcionalidad de los componentes. El resultado es que se incluyen características que están "de más", provocando una sobrecarga de recursos que se hace más grande en cuanto más amplio es el campo de reutilización.

El término framework tiene una acepción más amplia, en donde además de incluir una biblioteca de componentes reutilizables, es toda una tecnología o modelo de programación que contiene máquinas virtuales, compiladores, bibliotecas de administración de recursos en tiempo de ejecución y especificaciones de lenguajes.

Symfony: Es un framework PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional, de tipo full-stack construido con varios componentes independientes creados por el proyecto Symfony. Es un completo Framework diseñado para optimizar, gracias a sus características, el desarrollo de las Aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.[22]

Sus principales características son:[22]

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

Capítulo I: Fundamentación teórica

- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Bootstrap: Es un framework CSS desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. Es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías.[23]

Como principales características tiene:[23]

- Soporte bastante bueno (casi completo) con HTML5 y CSS3, permitiendo ser usado de forma muy flexible para desarrollo web con unos excelentes resultados.
- Se ha añadido un sistema GRID que permite diseñar usando un GRID de 12 columnas donde se debe plasmar el contenido, con esto podemos desarrollar responsive de forma mucho más fácil e intuitiva.
- Establece Media Queries para 4 tamaños de dispositivos diferentes variando dependiendo del tamaño de su pantalla, estas Media Queries permiten desarrollar para dispositivos móviles y tablets de forma mucho más fácil.
- También permite insertar imágenes responsive, es decir, con solo insertar la imagen con la clase “img-responsive” las imágenes se adaptarán al tamaño.
- Además, es compatible con casi todos los navegadores web existentes.

Todas estas características hacen que Bootstrap sea una excelente opción para desarrollar webs y aplicaciones web totalmente adaptables a cualquier tipo de dispositivo.

JQuery: Es un framework JavaScript. Pues es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para

Capítulo I: Fundamentación teórica

realizar tareas habituales. Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc.[24]

1.12 Herramientas utilizadas.

Visual Paradigm para UML: Es una herramienta para desarrollo de aplicaciones utilizando modelado UML* ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.[15]

Netbeans: Es un entorno de desarrollo gratuito y de código abierto que en el momento de escribir este artículo está en su versión 7.4. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP, Groovy, C/C++, HTML5, entre otros. Además, puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS.[25]

1.13 Conclusiones del capítulo.

En este capítulo se definieron los conceptos asociados al dominio del sistema, además de realizar un estudio de los softwares existentes relacionados con la gestión de la información referente al comportamiento de las plagas y enfermedades de los cultivos. Se analizaron además los diferentes tipos metodologías de desarrollo de software, utilizando RUP como metodología de desarrollo. Se dieron a conocer las tecnologías utilizadas en la realización de la aplicación, como son PHP, HTML, CSS, para la programación y diseño. Se utilizó MySQL como sistema gestor de base de datos, como servidor Web Apache ya que este es gratuito y con la característica de ser multiplataforma, y como Framework para el desarrollo de la aplicación Symfony.

Capítulo II: Construcción del Sistema Informático Propuesto

Capítulo II: Construcción del Sistema Informático Propuesto

Introducción al capítulo.

Se describen las características y conceptos fundamentales en los procesos del negocio, definiendo los actores y trabajadores que interactúan con el mismo, se define el diagrama de casos de uso del negocio, los requisitos funcionales y no funcionales, se realiza también un diagrama de casos de uso del sistema y se describen los mimos. Se generan los diagramas de clases del diseño, y diagrama físico y lógico de datos, los diagramas de clases web además del diagrama de despliegue.

2.1 Breve descripción de los procesos del negocio.

Los Departamentos de Sanidad Vegetal en Cuba, tienen como misión, contribuir a la prevención y disminución de los riesgos y pérdidas por plagas sin afectar al ambiente sobre una base sostenible. Para lograr esto, los mismos necesitan nutrirse de información recopilada por otras entidades como las Estaciones de Protección de Plantas (EPP), que mensualmente hace un estudio sobre el desarrollo de los cultivos en los campos y las plagas y enfermedades que puedan presentarse, así como las incidencias de las mismas sobre los cultivos, estainformación se clasifica (después de aprobar su corrección), para construir los modelos, a partir de los cuales se toman las decisiones para erradicar los diversos tipos de plagas que se hayan presentado, y se muestran tablas y gráficas con datos actuales del estado de los cultivos, con los cuales se pueden hacer diferentes cálculos estadísticos.

2.2 Reglas del negocio.

- El consultor del departamento de sanidad vegetal deberá ser el único encargado de recibir la información recopilada por los analistas, así como de confeccionar los modelos para la toma de decisiones.
- Para confeccionar los diferentes tipos de modelos, se deberá revisar la información obtenida.
- El director provincial será quien tome las decisiones.

Capítulo II: Construcción del Sistema Informático Propuesto

2.3 Modelo de casos de uso del negocio.

El modelo de casos de uso del negocio se utiliza para conseguir un acuerdo con los usuarios y clientes sobre qué debería hacer el sistema para los usuarios, siendo así una especificación completa de todas las formas posibles para utilizar el sistema.

El modelo de casos de uso del negocio describe los procesos de una empresa en términos de casos de uso y actores del negocio en correspondencia con los procesos del negocio y los clientes, respectivamente.[26]

2.3.1 Actores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actores del negocio

Nombre del Actor	Descripción
Director provincial	Es el encargado de solicitar al consultor de la provincia el informe de control de plagas para la toma de decisiones.

Tabla 1: Actores del negocio

2.3.2 Trabajadores del negocio.

Un trabajador del negocio representa a un individuo, software o hardware que desempeña un rol dentro de las realizaciones de los casos de uso del negocio.

Tabla 2: Trabajadores del negocio

Nombre del Trabajador	Descripción
Analista	Es el encargado de obtener la información necesaria en las estaciones de protección, almacenarla y enviarla al consultor.

Capítulo II: Construcción del Sistema Informático Propuesto

<p>Consultor</p>	<p>Es el encargado de recibir la información obtenida por los analistas, clasificarla, confeccionar los diferentes tipos de modelos y finalmente entregar un informe al director provincial para la aprobación de decisiones tomadas.</p>
------------------	---

2.3.3 Diagrama de casos de uso del negocio.

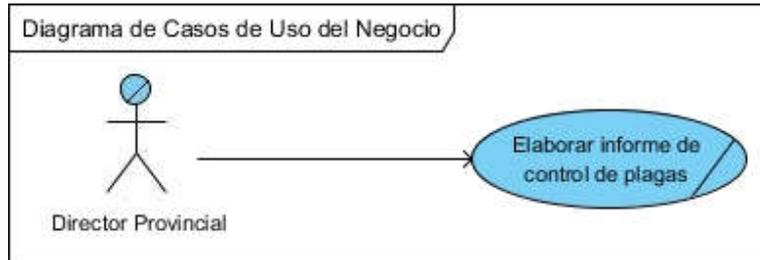
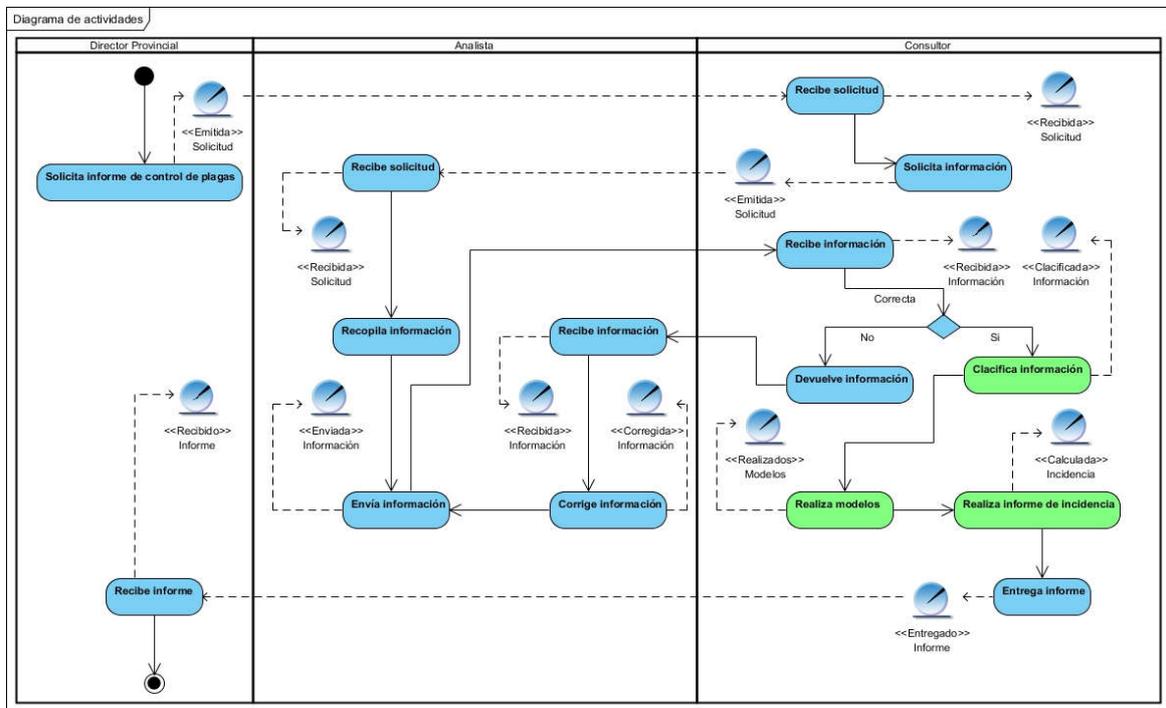


Figura 1: Diagrama de Casos de Uso del Negocio.

2.3.4 Descripciones de los casos de uso del negocio.

En este epígrafe se describe detalladamente los casos de uso del negocio donde se evidencia la interacción entre los trabajadores y actores del negocio, para ello se utiliza un diagrama de actividades del caso de uso identificado, el cual se muestra a continuación.



Capítulo II: Construcción del Sistema Informático Propuesto

Figura 2: Diagrama de actividades del caso de uso del Obtener Información.

2.4 Modelo de objeto del negocio.

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo.[27]

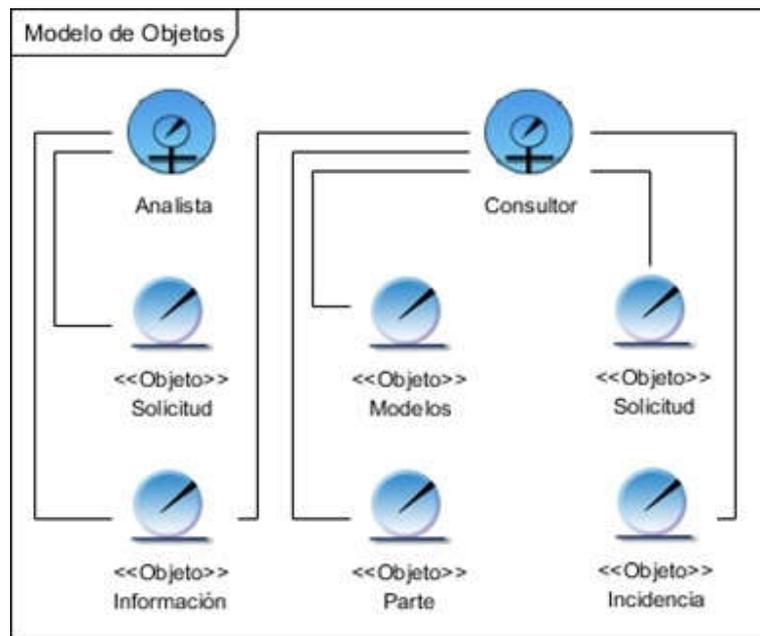


Figura 3: Diagrama de clases del modelo de objetos del negocio.

2.5 Descripción del modelo del sistema.

Desde el momento en que se haya puesto en marcha el sistema realizado, se podrá insertar la información en el mismo desde cualquier puesto de trabajo perteneciente a la empresa.

2.5.1 Requerimientos funcionales.

Expresan la naturaleza del funcionamiento del sistema (como interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento).[28]

1. Autenticarse
2. Cerrar Sesión

Capítulo II: Construcción del Sistema Informático Propuesto

3. Cambiar Contraseña
4. Crear Usuario
5. Insertar Usuario
6. Modificar Usuario
7. Eliminar Usuario
8. Buscar Usuario
9. Listar Municipio
10. Crear Municipio
11. Modificar Municipio
12. Eliminar Municipio
13. Listar Empresa
14. Crear Empresa
15. Modificar Empresa
16. Eliminar Empresa
17. Listar Rama
18. Crear Rama
19. Modificar Rama
20. Eliminar Rama
21. Listar Unidad
22. Crear Unidad
23. Modificar Unidad
24. Eliminar Unidad
25. Listar Estación
26. Crear Estación
27. Modificar Estación
28. Eliminar Estación
29. Listar Cultivo
30. Crear Cultivo
31. Modificar Cultivo
32. Eliminar Cultivo
33. Listar Producto

Capítulo II: Construcción del Sistema Informático Propuesto

34. Crear Producto
35. Modificar Producto
36. Eliminar Producto
37. Listar Agente Nocivo
38. Crear Agente Nocivo
39. Modificar Agente Nocivo
40. Eliminar Agente Nocivo
41. Listar Modelo
42. Crear Modelo
43. Modificar Modelo
44. Eliminar Modelo
45. Buscar Modelo
46. Filtrar Modelo 2004
47. Exportar a PDF Modelo 2004
48. Filtrar Modelo 2004d
49. Exportar a PDF Modelo 2004d
50. Filtrar Modelo 2003
51. Exportar a PDF Modelo 2003
52. Visualizar por fecha

2.5.2 Requerimientos no funcionales.

Los requisitos no funcionales especifican los criterios que se deben usar para juzgar el funcionamiento de un sistema, en lugar de un comportamiento específico. En general, o sea son restricciones sobre el espacio y posibles soluciones.[28]

- **Apariencia o interfaz externa:** Debido a que el sistema trata con información a la Sanidad Vegetal en la provincia, por lo que se decide usar colores verde claro, carmelita y azul claro que referencien a los árboles, (Medio Ambiente. Siendo así fácil y asequible en cuanto a navegación refiérase.
- **Usabilidad:** El sistema será utilizado por los trabajadores del Departamento de Sanidad Vegetal, los cuales tendrán acceso a determinada información del mismo según los privilegios que este posea según su cargo en la entidad.

Capítulo II: Construcción del Sistema Informático Propuesto

- **Soporte:**Una vez se haya concluido el sistema, este se deberá someter a una serie de pruebas, las cuales tendrán el fin de determinar posibles fallas antes de que este sea utilizado por los usuarios finales.
- **Seguridad:**Se preverá un control estricto sobre la seguridad de la información según los privilegios de los usuarios, pues el control de los mismos partirá de una autenticación inicial estableciendo así un nivel de acceso a la información.
- **Software:**Se necesitará una versión igual o superior a Microsoft Windows XP. Para el servidor Webse necesitará Apache Http Server 2.2.22. Para la configuración del framework **Symfony** se requiere:
 - Versión de PHP 5.3.3 como mínimo.
 - Las librerías (vendor) de Symfony deben estar instaladas.
 - Los directorios app/cache y app/logs deben tener permisos de escritura.
 - Debe estar configurada "date.timezone" en el sistema.
 - Las funciones Json_encode(), ctype_alpha(), sesion_start(), deben estar disponibles.
 - Detect_unicode debe estar deshabilitado en el php.ini.

El servidor que actuará como soporte de datos deberá disponer de MySQL.

Del Lado del Cliente el servidor necesitará un navegador, preferentemente Mozilla Firefox 12, Google Chrome 18, Internet Explorer 9, o versiones más actualizadas.

- Hardware.

Se necesitará una máquina que se desempeñe como servidor con características como:

- Conectividad: 10MB/s.
- Espacio en disco disponible 10GB (mínimo).
- Procesador Pentium IV (mínimo).
- Memoria RAM 512Mb (mínimo).
- Una tarjeta de Red.

Capítulo II: Construcción del Sistema Informático Propuesto

2.6 Modelo de Caso de Uso del Sistema.

El modelo de casos de uso es la técnica más efectiva y a la vez la más simple que emplean los desarrolladores de software para modelar los requisitos del sistema desde la perspectiva del usuario. Consiste en actores (que representan usuarios y otros sistemas que interactúan con el software) y casos de uso (que representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor).[26]

A continuación, se muestran los casos de uso que se han definido. Primeramente, se muestra un modelo con la jerarquía de los actores atendiendo a sus privilegios en cuanto a nivel de acceso refiérase.

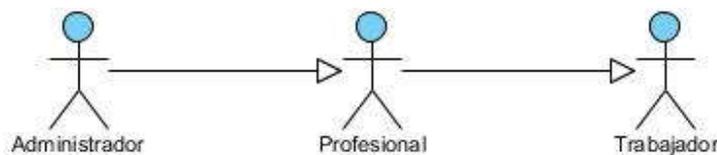


Figura 4: Jerarquía de los Actores.

2.6.1 Actores del Sistema.

Nombre del actor	Descripción
Administrador	Es la persona que tiene control total sobre el sistema, gestiona los usuarios del mismo y les otorga los niveles de acceso a la información que pueden consultar
Profesional	Persona que puede gestionar y consultar la información en el sistema.
Usuario	Persona que puede consultar la información del sistema. No puede modificar ninguna información.

Tabla 1: Descripción de los actores del sistema a automatizar.

2.6.2 Casos de uso del sistema.

Caso de Uso	Usuario	Requerimiento
Autenticarse	Anexo A.1	1
Cerrar sesión	Anexo A.2	2
Cambiar Contraseña	Anexo A.3	3

Capítulo II: Construcción del Sistema Informático Propuesto

Gestionar Usuario	Anexo A.4	4-8
Gestionar Municipio	Anexo A.5	9-12
Gestionar Empresa	Anexo A.6	13-16
Gestionar Rama	Anexo A.7	17-20
Gestionar Unidad	Anexo A.8	21-24
Gestionar Estación	Anexo A.9	25-28
Gestionar Campo	Anexo A.10	29-32
Gestionar Cultivo	Anexo A.11	33-36
Gestionar Producto	Anexo A.12	37-40
Gestionar Agente Nocivo	Anexo A.13	41-45
Gestionar Modelo	Anexo A.14	46-47
Visualizar Modelo 2004	Anexo A.15	48-49
Visualizar Modelo 2004d	Anexo A.16	50-51
Visualizar Modelo 2003	Anexo A.17	52

Tabla 4: Casos de uso del sistema.

2.6.3 Diagrama de Casos de Uso del Sistema.

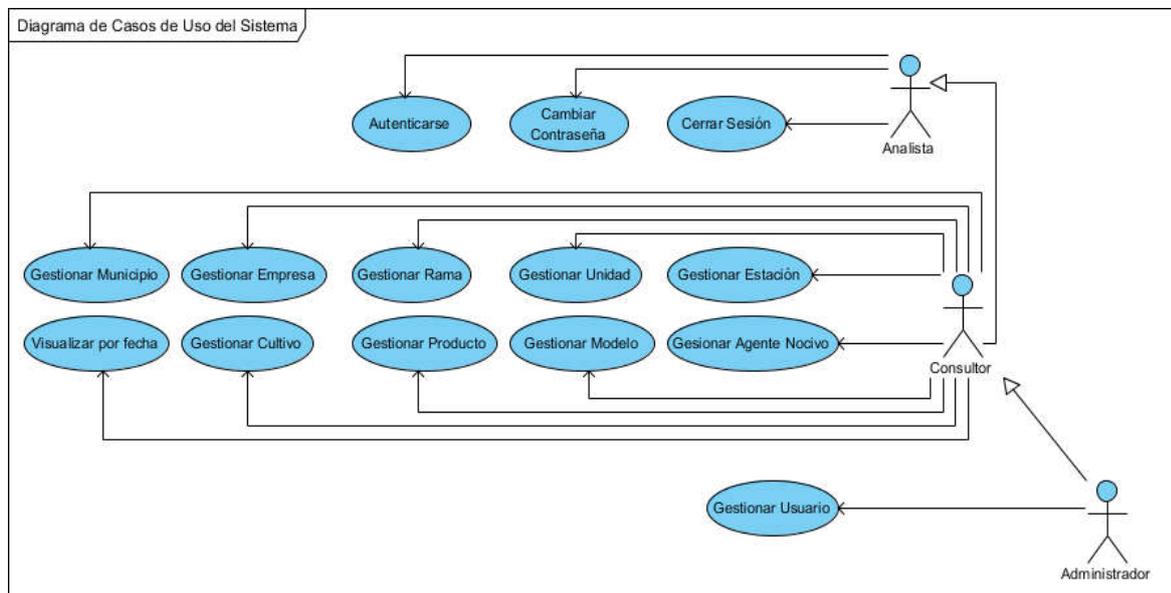


Figura 5: Diagrama de Casos de Uso del Sistema

Capítulo II: Construcción del Sistema Informático Propuesto

2.6.4 Descripción de los casos de uso del sistema

Caso de Uso	Descripción	Prototipo
Autenticarse	Anexo A.1	Anexo P.1
Cerrar sesión	Anexo A.2	Anexo P.2
Cambiar Contraseña	Anexo A.3	Anexo P.3
Gestionar Usuario	Anexo A.4	Anexo P.4
Gestionar Municipio	Anexo A.5	Anexo P.5
Gestionar Empresa	Anexo A.6	Anexo P.6
Gestionar Rama	Anexo A.7	Anexo P.7
Gestionar Unidad	Anexo A.8	Anexo P.8
Gestionar Estación	Anexo A.9	Anexo P.9
Gestionar Cultivo	Anexo A.10	Anexo P.10
Gestionar Producto	Anexo A.11	Anexo B.11
Gestionar Agente Nocivo	Anexo A.12	Anexo P.12
Gestionar Modelos	Anexo A.13	Anexo P.13
Visualizar Modelo 2003	Anexo A.14	Anexo P.14
Visualizar Modelo 2004	Anexo A.15	Anexo P.15
Visualizar Modelo 2004D	Anexo A.16	Anexo P.16
Visualizar por fecha	Anexo A.17	Anexo P.17

Tabla 5: Descripción de los casos de uso del sistema.

2.7 Construcción del Sistema.

2.7.1 Diagramas de clases del diseño.

Caso de Uso	Prototipo
Autenticarse	Anexo C.1
Cerrar sesión	Anexo C.2
Cambiar Contraseña	Anexo C.3
Gestionar Usuario	Anexo C.4
Gestionar Municipio	Anexo C.5
Gestionar Empresa	Anexo C.6
Gestionar Rama	Anexo C.7
Gestionar Unidad	Anexo C.8

Capítulo II: Construcción del Sistema Informático Propuesto

Gestionar Estación	Anexo C.9
Gestionar Cultivo	Anexo C.10
Gestionar Producto	Anexo C.11
Gestionar Agente Nocivo	Anexo C.12
Gestionar Modelos	Anexo C.13
Visualizar Modelo 2003	Anexo C.14
Visualizar Modelo 2004	Anexo C.15
Visualizar Modelo 2004D	Anexo C.16
Visualizar por fecha	Anexo C.17

Tabla 6: Diagramas de clases del diseño.

2.8 Diseño de la Base de Datos.

En el desarrollo de cualquier sistema informático resulta de vital importancia realizar un correcto diseño de la base de datos. A continuación, se muestra el diseño de la misma, partiendo de la descripción de un modelo lógico y físico.

Una base de datos (BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible, utilizados por programas y usuarios. Por lo tanto, el concepto de BD generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término **base**. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.[29]

2.8.1 Modelo lógico de datos.

El modelo lógico de datos consiste en traducir el esquema de datos conceptual en un esquema de datos lógico para un DBMS específico y obtener una representación del modelo conceptual que use de forma eficiente las facilidades de estructuración de datos y modelado de restricciones, disponibles en el modelo.[30]

Capítulo II: Construcción del Sistema Informático Propuesto

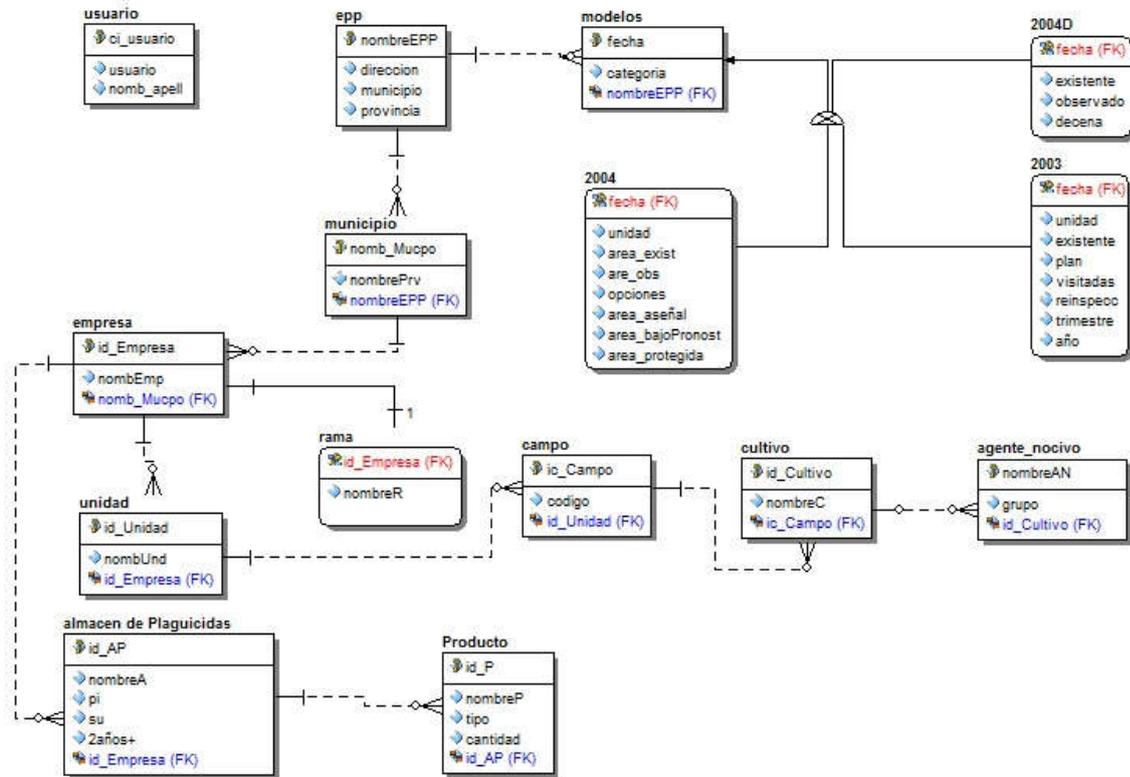


Figura 6: Modelo Lógico de Datos

2.8.2 Modelo físico de datos.

El modelo físico de datos consiste en diseñar el esquema físico a través del refinamiento del esquema lógico, para su implementación en un DBMS específico e Implementar el modelo lógico empleando de forma eficiente las estructuras físicas del DBMS, de forma de obtener optimizar la performance del sistema.[31]

Capítulo II: Construcción del Sistema Informático Propuesto

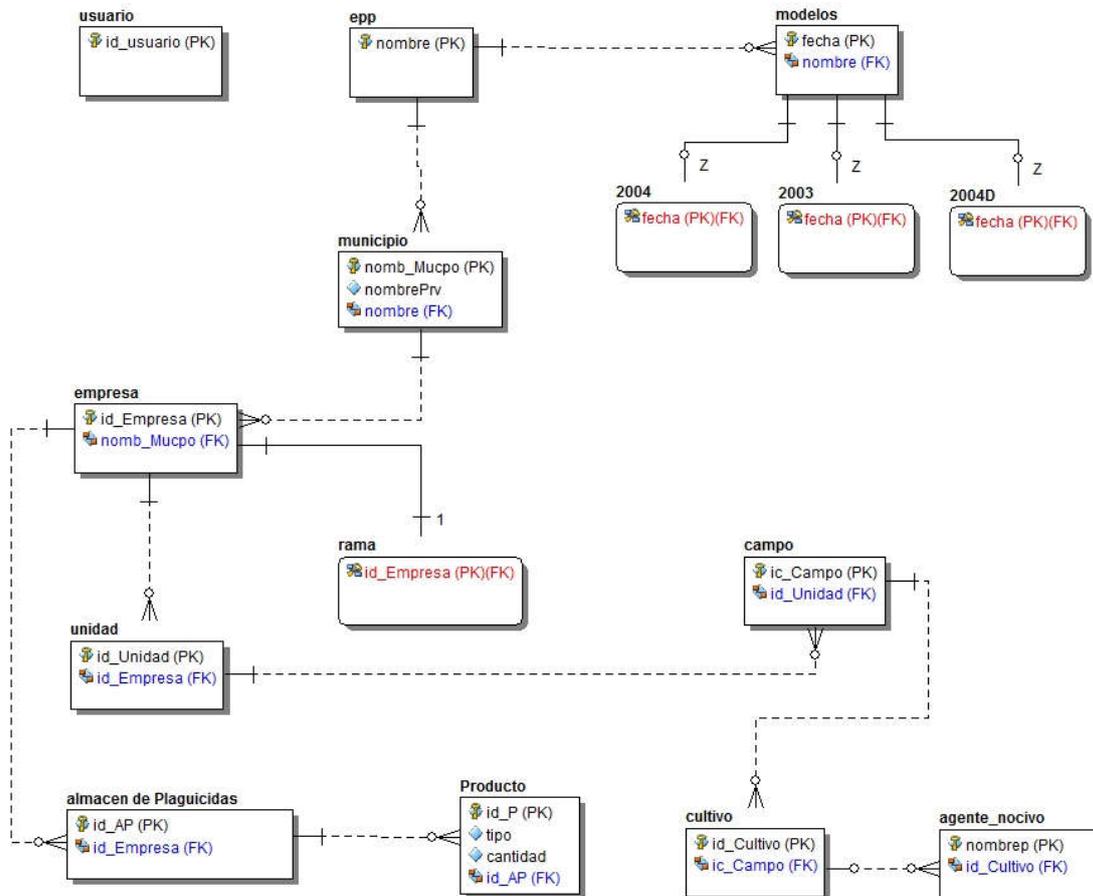


Figura 7: Modelo Físico de Datos.

2.9 Modelo de despliegue.

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP.[32]

Capítulo II: Construcción del Sistema Informático Propuesto

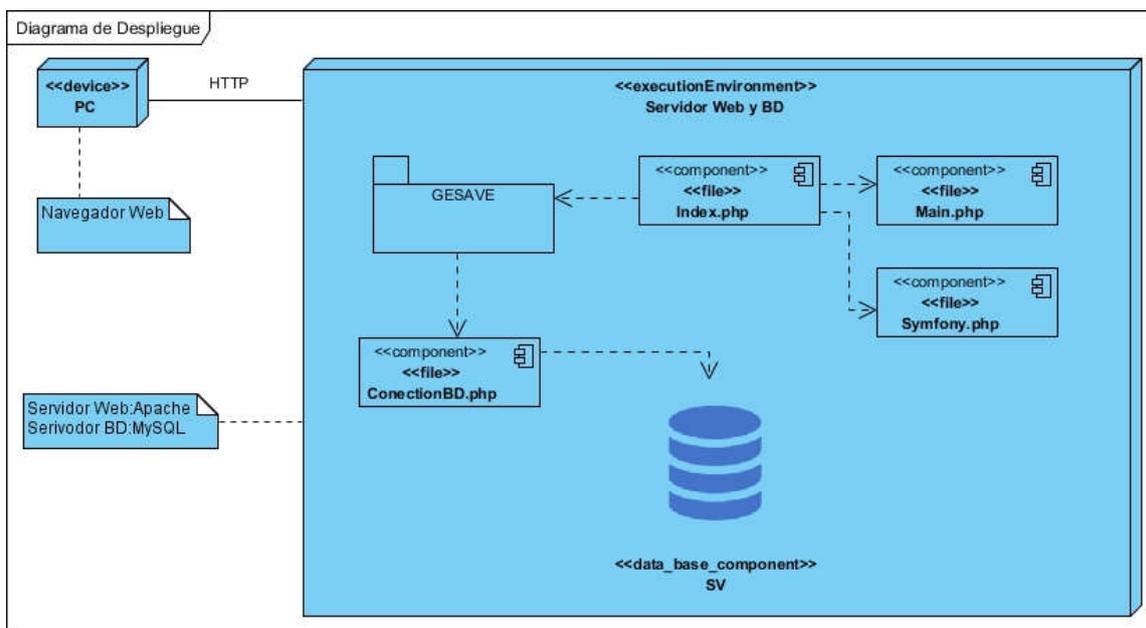


Figura 8: Modelo de despliegue.

2.9 Principios de diseño.

El diseño de sistemas es definido como el proceso de aplicar ciertas técnicas y principios con el fin de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

2.9.1 Interfaz de usuario.

Para la construcción de la interfaz del sistema, se aprovecharon las ventajas que ofrece Bootstrap, este se encarga de la tipografía, formularios, botones, tablas e íconos. La interacción con el mismo se basa en selecciones de tipo menú, así como acciones físicas con elementos de código visual llamados botones. Para el diseño del sistema se emplearon colores verdes y azules en varias tonalidades para los menús, y predominando el blanco para los formularios. Cuenta con un equilibrio para evitar la recarga visual distribuyéndola de manera cómoda cumpliendo con la regla de distribución de la atención: de izquierda a derecha y de arriba hacia abajo. Para los textos se utilizaron fuentes de tipo Arial de varios tamaños dependiendo del contexto.

Capítulo II: Construcción del Sistema Informático Propuesto

2.9.2 Tratamiento de excepciones.

En todo software, las excepciones son problemas que se presentan a menudo que afectan el correcto funcionamiento de los mismos en el tiempo de su ejecución. El sistema propuesto está diseñado de forma tal que las posibilidades de que los usuarios introduzcan información errónea sean mínimas. Pues, aunque en la mayoría de las ocasiones el usuario teclea y también selecciona información de los elementos en pantalla, es necesario mantener las técnicas de validación de datos ofrecidas por Bootstrap, el cual informa los errores de datos mediante mensajes de error, de fácil comprensión y visualización para los usuarios.

2.10 Conclusiones del capítulo.

En este capítulo se describieron los procesos que se realizan en el Departamento de Sanidad Vegetal. El modelado del sistema se realizó mediante la captura de los requerimientos funcionales y no funcionales, así como la identificación y descripción de los actores y trabajadores del sistema, necesarios en el desarrollo del sistema propuesto.

Se realizaron los modelos lógico y físico de la base de datos, modelo de despliegue y diagramas de clases web y los principios de diseño del sistema, lo cual permitió hacer un análisis completo y detallado del modelo del sistema y creando una guía para la implementación.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales.

Introducción al capítulo.

En este capítulo se realiza un análisis de los elementos necesarios para determinar los costos y beneficios que implica el desarrollo del sistema propuesto. El análisis se realiza mediante el método de Estimación de análisis de Puntos de Casos de Usos. Finalmente se presenta la validación del sistema mediante los Casos de Prueba Funcionales.

3.1 Estudio de factibilidad.

El estudio de factibilidad es un análisis que se realiza antes del equipo comprometerse con la realización del software. La técnica que se usa es el Análisis de Puntos de Casos de Uso. Este realiza una comparación entre los costos y beneficios del sistema informático a desarrollar, permitiendo determinar si es fiable o no la realización del mismo.

3.1.1 Puntos de casos de uso sin ajustar.

Se calcula a partir de la siguiente ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Dónde:

- **UUCP**: Puntos de casos de uso sin ajustar.
- **UAW**: Factor de peso de los actores sin ajustar.
- **UUCW**: Factor de peso de los casos de uso sin ajustar.

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y, en segundo lugar, la forma en la que el actor interactúa con el sistema.

Tipo de Actor	Descripción	Factor de Peso
Simple	Otro sistema interactúa con el sistema a desarrollar mediante una interfaz de programación (API).	1

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

Medio	Otro sistema interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 7: Tipo de Actor, Descripción y su Factor de Peso.

Actor	Tipo de Actor	Factor de Peso
Administrador	Complejo	3
Profesional	Complejo	3
Usuario	Complejo	3

Tabla 8: Especificación de los tipos de actores del sistema.

Al tener 3 actores a interactuar con el sistema, los cuales son personas que interactúan con el sistema mediante una interfaz gráfica, se clasifican en actores de tipo Complejo, por tanto, su peso es 3:

Luego el factor de actores sin ajustar se calcula mediante la siguiente ecuación:

$$UAW = (\text{Cantidad de Actores}) * \text{Peso}$$

$$UAW = 3 * 3 = 9$$

Factor de casos de uso sin ajustar:

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones que se efectúan en ellos, donde una transacción se entiende como una secuencia de actividades atómica; es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Tipo de caso de uso	Descripción	Factor de Peso
Simple	El caso de uso contiene de 1 a 3 transacciones.	5
Medio	El caso de uso contiene de 4 a 7 transacciones.	10
Complejo	El caso de uso contiene más de 8	15

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

	transacciones.	
--	----------------	--

Tabla 9: Descripción de tipos de casos de uso de acuerdo a su complejidad.

Caso de Uso	Clasificación
Autenticarse	Simple
Cerrar sesión	Simple
Cambiar Contraseña	Simple
Gestionar Usuario	Medio
Gestionar Municipio	Simple
Gestionar Empresa	Simple
Gestionar Rama	Simple
Gestionar Unidad	Simple
Gestionar Estación	Simple
Gestionar Campo	Simple
Gestionar Cultivo	Simple
Gestionar Agente Nocivo	Simple
Gestionar Producto	Simple
Gestionar Almacén	Simple
Gestionar Modelo	Medio
Visualizar Modelo 2004	Simple
Visualizar Modelo 2004d	Simple
Visualizar Modelo 2003	Simple
Graficar Modelo 2004	Simple

Tabla 10: Clasificación de los casos de uso del sistema.

Tipo de caso de uso	Cantidad de casos de uso	Factor de Peso por Caso de Uso	Peso Total
Simple	17	5	85
Medio	2	10	20

Tabla 11: Peso Total según Cantidad y Tipo de Caso de Uso.

En la tabla anterior se muestran 16 casos de uso de clasificación simple y 2 de clasificación media, por lo que se les asigna un peso por caso de uso de 5 y 10 respectivamente, por tanto, se tiene que el Factor de Peso por Caso de Uso resulta:

$$UUCW = 17 \cdot 5 + 2 \cdot 10 = 105$$

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

Ya obtenidos los valores de Factor de Peso de Actores y Casos de Uso sin Ajustar, podemos obtener los Puntos de Casos de Uso si Ajustar, resultando:

$$\mathbf{UUCP = UAW + UUCW}$$

$$\mathbf{UUCP = 9+105 = 114}$$

3.1.2 Puntos de Casos de Uso Ajustados.

Una vez obtenidos los Puntos de Casos de Uso sin Ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$\mathbf{UCP = UUCP * TCF * EF}$$

Dónde:

- **UCP:** Puntos de Casos de Uso Ajustados
- **UUCP:** Puntos de Casos de Uso sin Ajustar
- **TCF:** Factor de Complejidad Técnica
- **EF:** Factor de Ambiente

Factor de Complejidad Técnica (TCF):

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Factor	Descripción	Peso	Importancia	Peso Total
T1	Sistema distribuido.	2	2	4
T2	Tiempo de respuesta.	1	3	3
T3	Eficiencia del usuario final.	1	5	5
T4	Procesamiento interno complejo.	1	2	2
T5	El código debe ser reutilizable.	1	3	3
T6	Facilidad de instalación.	0,5	3	1,5
T7	Facilidad de uso.	0,5	2	1
T8	Portabilidad.	2	2	4
T9	Facilidad de cambio.	1	2	2
T10	Concurrencia.	1	2	2

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

T11	Incluye objetos especiales de seguridad.	1	1	1
T12	Provee acceso directo a terceras partes.	1	1	1
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	1

Tabla 12: Factor determinante de la complejidad técnica del sistema.

$$\sum (\text{Peso} \cdot \text{Valor}) = 4+3+5+2+3+1,5+1+4+2+2+1+1+1 = 48,5$$

Resultando Factor de complejidad Técnica:

$$\text{TCF} = 0,6 + 0,01 \cdot \sum (\text{Peso} \cdot \text{Valor})$$

$$\text{TCF} = 0,6 + 0,01 \cdot 48,5$$

$$\text{TCF} = 1,085$$

Factor Ambiente (FE):

Las habilidades y el entrenamiento del grupo de desarrollo involucrado son de vital importancia, siendo estos, factores que completan el cálculo del Factor Ambiente, el cálculo del mismo es similar al cálculo del Factor de Complejidad Técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Importancia	Peso Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	5	7,5
E2	Experiencia en la aplicación.	0.5	4	2
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2,5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	4	8
E7	Personal a tiempo parcial.	-1	3	-3
E8	Dificultad del lenguaje de programación.	-1	3	-3

Tabla 13: Factor Ambiente.

$$\sum (\text{Peso} \cdot \text{Valor}) = 2,5+2+4+2,5+5+8-3-3 = 23$$

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

$$EF=1,4 - 0,03*\Sigma(\text{Peso} * \text{Valor})$$

$$EF = 1,4 - 0,03 * 23$$

$$EF = 0,71$$

Resultando los Casos de Uso Ajustados:

$$\mathbf{UCP = UUCP * TCF * EF}$$

$$\mathbf{UCP = 114 * 1,085 * 0,71}$$

$$\mathbf{UCP = 87,819}$$

3.1.3 Estimación del esfuerzo.

Factores de ambiente por debajo de medio (E1-E6).	2
Factores de ambiente por debajo de medio (E7-E8).	0
Total	2
Facto de conversión(FC)(horas/hombre)	20

Tabla 14: Determinación del esfuerzo.

CF: Factor de Conversión.

CF: 20 horas/hombre

El esfuerzo en horas hombres se determina:

$$\mathbf{E = UCP * CF}$$

Dónde:

- **E:** Esfuerzo estimado en horas-hombre.
- **UCP:** Puntos de Casos de Uso Ajustados.
- **CF:** Factor de conversión.

Para determinar el Factor de Conversión (**CF**)

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

- Si el total es 2 o menos, se utiliza el **factor de conversión 20 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el **factor de conversión 28 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Como se muestra previamente de manera tabular se puede apreciar que el total es menor a 2, por lo que se utiliza el Factor de Conversión 20 horas/hombre. Resultando el esfuerzo para calcular los casos de uso del sistema:

$$E = UCP * CF$$

$$E = 87,819 * 20$$

$$E = 1756,38 \text{ horas/hombre}$$

Calculado el esfuerzo total para el desarrollo del sistema, se puede calcular a partir del valor obtenido, el tiempo necesario y a través de este el costo vinculado, así como los beneficios tangibles e intangibles que se obtienen con el desarrollo del sistema.

Duración:

Trabajando 30 días al mes y 8 horas como promedio, resulta:

$$\text{Duración(días)} = \text{Totaldehoras/hombreentre8horasaldía}$$

$$= 1756,38/8 = 219,54 \text{ días.}$$

$$\text{Duración (meses)} = \text{Total de días /30 días por mes} = 219,54 /30$$

$$= 7,31 \approx 7 \text{ meses.}$$

Actividad	Valor %	Esfuerzo (h/h)
Análisis	30	896,64
Diseño	20	597,76

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

Implementación	35	1046,08
Prueba	10	298,88
Sobrecarga	5	149,44
Total de Horas.	100	2834,8

Tabla 15: Criterio de distribución de esfuerzos.

3.1.4 Cálculo de costo.

Contando con un salario promedio de \$400,00:

Costo = 7 meses * \$400,00 mensual = \$2800,00 CUP

3.2 Análisis de los beneficios tangibles e intangibles.

Con el desarrollo del sistema informático propuesto como trabajo de diploma, los resultados son mayormente intangibles, pues tiene como objetivo facilitar el trabajo de las personas que laboran en el Departamento de Sanidad Vegetal perteneciente al MINAG. Aunque en conjunto con el desarrollo de cualquier producto informático va asociado un costo y su justificación está dada por los beneficios tangibles e intangibles. Para el desarrollo de este sistema no se necesitó invertir en medios técnicos, significando esto un ahorro considerable del tiempo invertido en la gestión y control de la información manejada.

3.3 Casos de Pruebas Funcionales.

Para entender que son los casos de pruebas funcionales, debemos conocer que son las pruebas funcionales.

Pruebas Funcionales:

Las pruebas funcionales son las encargadas de validar cuando el comportamiento observado del software probado cumple o no con sus especificaciones.[33]

Casos de Pruebas:

Los casos de prueba especifican los requisitos de la aplicación, por lo que cada requisito debe estar cubierto por un mínimo de un caso de prueba. Cada caso de prueba está compuesto por varios pasos a ejecutar, dependiendo de la complejidad del

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

caso, y cada paso está compuesto por una acción, que será realizada por el tester, y un resultado esperado. Para que un caso de prueba resulte exitoso, todos los pasos deben cumplir el resultado esperado. Si uno de los pasos no lo cumple, todo el caso de prueba resultará fallido.[34]

Paralelo se aplicó el método de casos de prueba funcional a los casos de uso del sistema propuesto, seleccionando aquellos con los que el usuario tiene mayor interacción (entrada de datos).

3.3.1 Caso de Prueba para Autenticarse.



Acción: Iniciar sesión.

Validaciones: Validando los siguientes campos en el evento "onClick" del botón *Ingresar*.

Nombre de usuario: Cadena de caracteres, que debe ser el nombre de un usuario registrado en la Base de Datos del sistema, esta no podrá incluir números ni caracteres especiales. Obligatorio (*)

Contraseña: Cadena de caracteres, que deberá tener al menos 8 caracteres, mayúsculas, minúsculas, números y caracteres especiales. Obligatorio (*)

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

De no ser cumplidos estos requisitos el sistema mostrará un mensaje de error con alguna de las siguientes informaciones:

- Por favor rellene este campo. (Ubicado debajo de cada uno de los formularios)
- Acceso Denegado, Usuario o Contraseña incorrecto. (Ubicado en la parte superior del formulario)
- Solo letras. (Ubicado debajo del formulario de Nombre y Apellidos)

3.3.2 Caso de Prueba para Cambiar contraseña.



El formulario para cambiar la contraseña está contenido en un recuadro gris claro. En la parte superior, el título "Contraseña" está en negrita. Debajo de él hay un campo de entrada rectangular con el texto "Contraseña" en gris. A continuación, el título "Repetir Contraseña" también está en negrita, seguido de otro campo de entrada rectangular con el texto "Contraseña" en gris. En la parte inferior del recuadro, hay un botón rectangular de color verde con el texto "Cambiar" en blanco.

Acción: Cambiar contraseña.

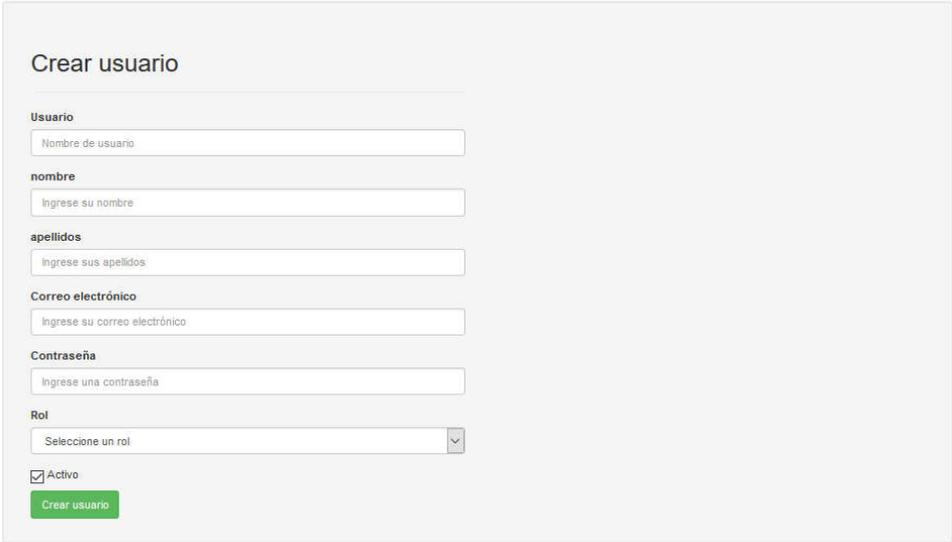
Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Contraseña y repetir contraseña se podrá insertar una cadena de caracteres de tipo *AlpaNumerico-Espacios*, de una longitud de 6 caracteres como mínimo, y estos deben coincidir de lo contrario el sistema mostrará el siguiente mensaje de error (*Mínimo 6 caracteres, Las contraseñas no coinciden*) respectivamente en la parte inferior del campo.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

En el evento del botón Cambiar, se muestra un mensaje de éxito o de error en la parte superior de la vista.

3.3.3 Caso de Prueba para Gestionar Usuario.



The screenshot shows a web application interface with a dark navigation bar at the top containing the following menu items: 'Backend Aplicación', 'Gestionar Usuario', 'Gestionar municipio', 'Gestionar empresa', 'Gestionar rama', 'Gestionar unidad', and 'Dropdown'. A search bar and a 'Submit' button are also present in the navigation bar. The main content area is titled 'Crear usuario' and contains the following form fields:

- Usuario:** A text input field labeled 'Nombre de usuario'.
- nombre:** A text input field labeled 'Ingrese su nombre'.
- apellidos:** A text input field labeled 'Ingrese sus apellidos'.
- Correo electrónico:** A text input field labeled 'Ingrese su correo electrónico'.
- Contraseña:** A text input field labeled 'Ingrese una contraseña'.
- Rol:** A dropdown menu labeled 'Seleccione un rol'.
- Activo**
- Crear usuario** button.

La gestión de los usuarios es de vital importancia pues solo podrá acceder al sistema aquel usuario que este registrado en la base de datos del sistema. Para ello se deben llenar los campos necesarios y luego presionar el botón *Crear usuario*.

Acción: Crear usuario.

Validaciones:

- Todos los campos son de carácter obligatorio, en caso que se omita alguno de estos el sistema mostrará el siguiente mensaje de error (*Por favor rellene este campo*), al lado del campo que se omite.
- En los campos: Nombre y Apellidos, solo podrá insertarse una cadena de caracteres la cualdeberán ser letras solamente, en caso de que ocurra lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del formulario.
- En el campo: Correo electrónico, solo se podrán insertar datos que concuerden con la estructura válida para direcciones de correo, de lo contrario se mostrarán

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

los siguientes mensajes de error (*Información no válida.*) en la parte inferior del campo.

- En el campo: Rol se deberá escoger un rol, de lo contrario el sistema mostrará el siguiente mensaje de error en la parte inferior del formulario(*Por seleccione un Rol*)

En el evento del botón *Crear usuario*, se muestra un mensaje de éxito o de error en la parte superior de la vista.

3.3.4 Caso de Prueba para Gestionar Modelo.

Los modelos permiten almacenar y visualizar de una manera concreta la información y estado actual de los datos recogidos de cada unidad y empresa respectivamente. Para esto se deben insertar los datos en los formularios pertinentes y luego presionar el botón *Crear Modelo*.

Acción: Crear Modelo.

Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Opciones, Área señalizada, área bajo pronóstico, área bajo inspección solo se podrán insertar números positivos, iguales o mayores que cero, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo números positivos*), en la parte inferior de cada campo.

En el evento del botón *Crear Modelo*, se muestra un mensaje de éxito o de error en la parte superior de la vista.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.5 Caso de Prueba para Gestionar Empresa



Crear empresa

Nombre empresa

Rama

Municipio

Create enterprise

Acción: Crear Empresa.

Validaciones:

- Es de carácter obligatorio llenar todos los campos mostrados.
- En el campo: Nombre Empresa, solo podrá insertarse una cadena de caracteres la cual deberán ser letras solamente, en caso de que ocurra lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del formulario.
- En los campos: Rama y Municipio, se deberá escoger una opción según corresponda, de lo contrario el sistema mostrará el siguiente mensaje de error (*Seleccione una rama o Seleccione un municipio respectivamente*), en la parte inferior de los formularios mencionados.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.6 Caso de Prueba para Gestionar Rama.



The screenshot shows a web form titled "Crear Rama". Below the title is a horizontal line. Underneath, the label "Nombre rama" is positioned above a text input field containing the word "Rama". Below the input field is a green button with the text "Crear Rama". At the bottom of the form, there is a blue link with a bullet point that says "Regresar a la lista".

Acción: Crear Rama.

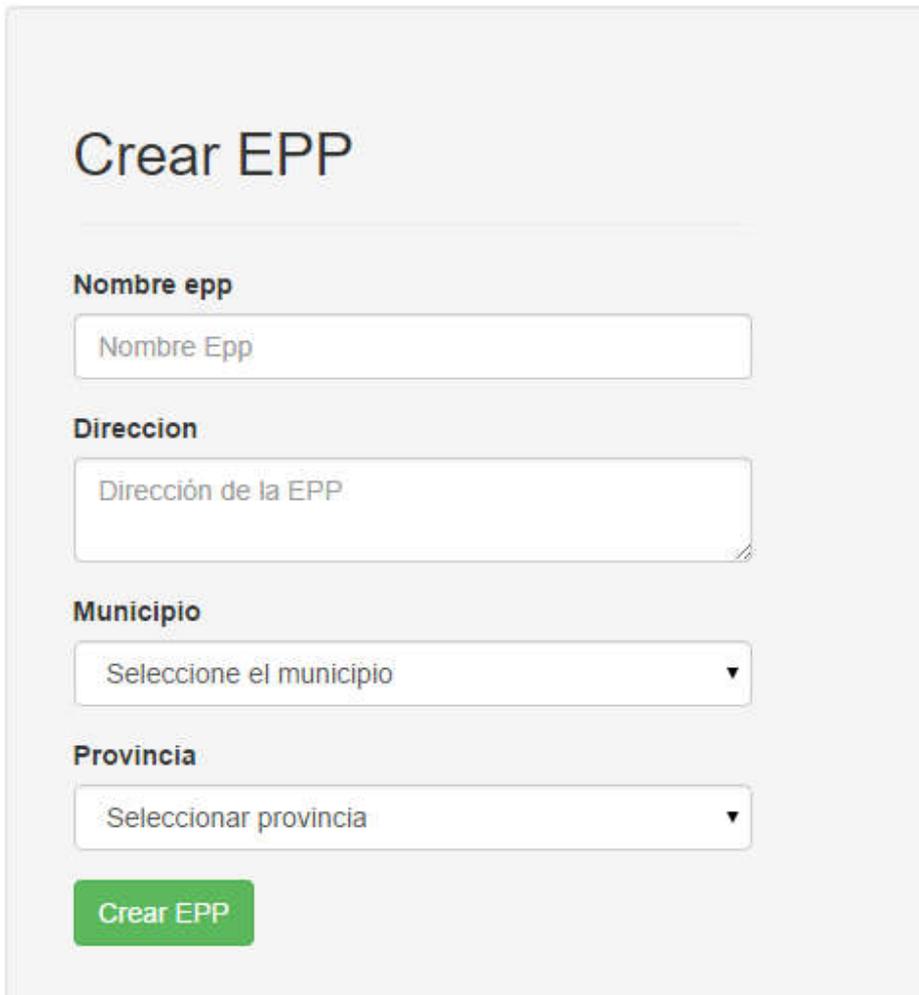
Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

En el evento del botón Crear rama, se muestra un mensaje de éxito o de error en la parte superior de la vista.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.7 Caso de Prueba para Gestionar EPP.



Crear EPP

Nombre epp

Direccion

Municipio

Provincia

Crear EPP

Acción: Crear EPP.

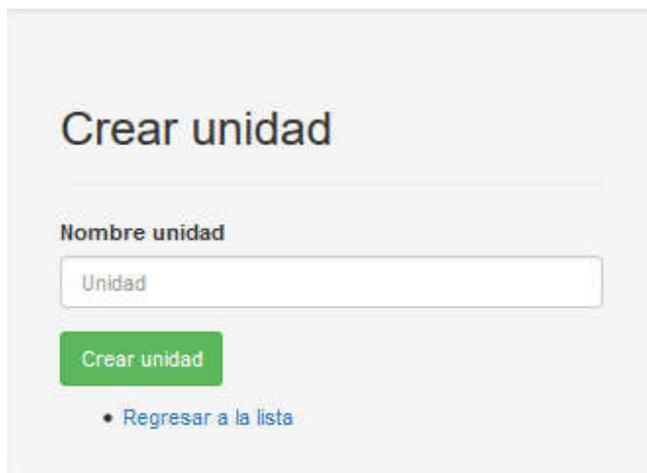
Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

En el evento del botón Crear EPP, se muestra un mensaje de éxito o de error en la parte superior de la vista.

3.3.8 Caso de Prueba para Gestionar Unidad.

La imagen muestra una interfaz de usuario con el título "Crear unidad". Debajo del título hay un campo de texto etiquetado "Nombre unidad" que contiene el texto "Unidad". Debajo del campo de texto hay un botón verde con el texto "Crear unidad". Debajo del botón hay un enlace azul con el texto "Regresar a la lista".

Crear unidad

Nombre unidad

Unidad

Crear unidad

• [Regresar a la lista](#)

Acción: Crear Unidad.

Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

En el evento del botón Crear Unidad, se muestra un mensaje de éxito o de error en la parte superior de la vista.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.9 Caso de Prueba para Gestionar Municipio.



Crear municipio

Nombre municipio

Provincia

Crear municipio

- [Regresar a la lista](#)

Acción: Crear Municipio.

Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

En el evento del botón Crear Municipio, se muestra un mensaje de éxito o de error en la parte superior de la vista.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.10 Caso de Prueba para Gestionar Cultivo.



Crear Cultivo

Nombre cultivo

Grupo cultivo

Seleccione un grupo ▼

Campo cultivo

Seleccione el campo ▼

[Crear cultivo](#)

- [Regresar a la lista](#)

Acción: Crear Cultivo.

Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

En el evento del botón Crear cultivo, se muestra un mensaje de éxito o de error en la parte superior de la vista.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.11 Caso de Prueba para Gestionar Agente Nocivo.



Crear plaga

Nombre

Crear plaga

- [Regresar a la lista](#)

Acción: Crear Agente Nocivo.

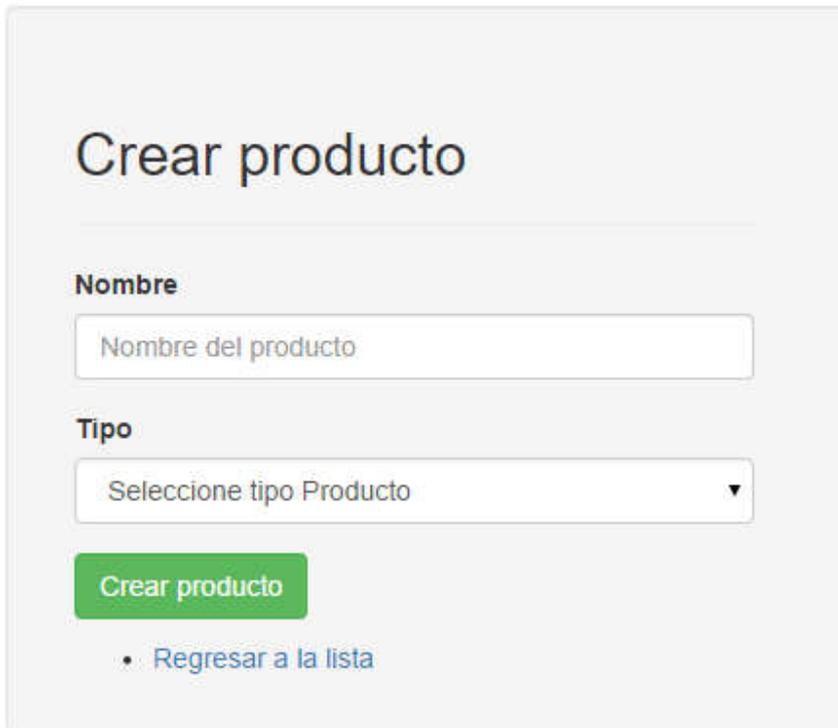
Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

En el evento del botón Crear Plaga, se muestra un mensaje de éxito o de error en la parte superior de la vista.

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

3.3.12 Caso de Prueba para Gestionar Producto.



Crear producto

Nombre

Tipo

Crear producto

- [Regresar a la lista](#)

Acción: Crear Producto.

Validaciones:

- Todos los campos son de carácter obligatorio, de lo contrario el sistema mostrará en la parte inferior el siguiente mensaje de error (*Por favor rellene este campo*).
- En los campos: Nombre, solo se podrá insertar una cadena de caracteres de tipo letra, de lo contrario el sistema mostrará el siguiente mensaje de error (*Solo letras*), en la parte inferior del campo.

En el evento del botón Crear Producto, se muestra un mensaje de éxito o de error en la parte superior de la vista.

-

Capítulo III: Estudio de factibilidad y casos de pruebas funcionales

Conclusiones del capítulo.

En este capítulo se realizó el análisis de factibilidad del producto de software propuesto, a partir del cual se estimó un tiempo de desarrollo de 7 meses y un costo de \$2800.00 CUP aproximadamente. Además, se determinaron los beneficios tangibles e intangibles concluyendo que es factible el desarrollo del sistema informático. Se realizaron además los casos de pruebas funcionales verificando el comportamiento del sistema y garantizando el control de la calidad en el desarrollo del software.

Conclusiones.

Teniendo en cuenta los objetivos planteados, se arriba a las siguientes conclusiones:

- ✓ Se realizó un análisis y un estudio del proceso de la información del departamento de Sanidad Vegetal en la provincia de Cienfuegos, identificando las particularidades del mismo.
- ✓ La metodología y herramientas utilizadas en el desarrollo del sistema se eligieron en aras de asegurar un correcto aprovechamiento del tiempo además de generar una documentación detallada del desarrollo del sistema propuesto.
- ✓ Se diseñó e implementó un sistema informático con funcionalidades que se adecuan a las necesidades de la institución, proporcionándole una herramienta que reduce el tiempo empleado en el proceso de control y utilización de la información de las plagas y enfermedades en los cultivos, dando cumplimiento de esta forma a los requerimientos planteados.
- ✓ Se realizaron casos de pruebas funcionales verificando el comportamiento del sistema, asegurando un correcto control de la calidad en el desarrollo del software, además de un estudio de factibilidad mediante el método de Estimación de Puntos de Casos de Usos, concluyendo que resultó factible el desarrollo del sistema propuesto.

Recomendaciones.

Apesar de que la investigación realizada cumplió con los objetivos trazados, se recomienda:

Recomendaciones

- ✓ Someter el sistema informático realizado a un período de tiempo significativo de pruebas, el cual se estime conveniente, con el fin de evaluar de manera práctica el desarrollo de sus funcionalidades detectando datos para su mejoramiento.
- ✓ Perfeccionar el sistema informático realizado, con la incorporación de nuevos módulos que tributen al enriquecimiento de los resultados obtenidos.
- ✓ Presentar el trabajo realizado en eventos científicos.
- ✓ Extender su utilización en los Departamentos de Sanidad Vegetal en otras provincias.

Referencias bibliográficas.

- [1] R. Santana, “¿Qué son las TIC?” [Online]. Available: <http://tutorial.cch.unam.mx/bloque4/lasTIC>. [Accessed: 12-Jun-2018].
- [2] P. Roque, “Plagas y enfermedades de las plantas: FAO in Emergencias.” [Online]. Available: <http://www.fao.org/emergencias/tipos-de-peligros-y-de-emergencias/plagas-y-enfermedades-de-las-plantas/es/>. [Accessed: 12-Jun-2018].
- [3] M. García, “Cultivo - EcuRed.” [Online]. Available: <https://www.ecured.cu/Cultivo>. [Accessed: 12-Jun-2018].
- [4] A. Castro, “Vigilancia Fitosanitaria.” [Online]. Available: <https://www.senasa.gob.pe/senasa/vigilancia-fitosanitaria/>. [Accessed: 12-Jun-2018].
- [5] “SAVIA. Un sistema web de ayuda a la... (PDF Download Available).” [Online]. Available: https://www.researchgate.net/publication/295519751_SAVIA_Un_sistema_web_de_ayuda_a_la_decision_para_control_de_plagas. [Accessed: 12-Jun-2018].
- [6] “Una ‘app’ para los agricultores y ganaderos | Agronews Castilla y León.” [Online]. Available: <http://www.agronewscastillayleon.com/una-app-para-los-agricultores-y-ganaderos>. [Accessed: 12-Jun-2018].
- [7] “Metodologías del Desarrollo de Software.” [Online]. Available: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>. [Accessed: 12-Jun-2018].
- [8] “MetodoPesadasRUP.pdf.” [Online]. Available: <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesRUP.pdf>. [Accessed: 09-May-2016].
- [9] “Arquitectura cliente/servidor.” [Online]. Available: <http://isa.uniovi.es/domotica/Temas/T6/T6-ClienteServidor.htm>. [Accessed: 12-Jun-2018].
- [10] “Definición arquitectura cliente servidor - Monografias.com,” 09-May-2016. [Online]. Available: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>. [Accessed: 09-May-2016].
- [11] “Tips de Desarrollo Web | Arquitectura N-Capas y LinqToSQL.” [Online]. Available: <http://www.nicholls.co/blog/post/Arquitectura-N-Capas-y-LinqToSQL>. [Accessed: 12-Jun-2018].
- [12] “IBM Knowledge Center - Patrón de diseño de modelo-vista-controlador.” [Online]. Available:

Referencias Bibliográficas

- https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm. [Accessed: 12-Jun-2018].
- [13] “¿Qué es un Sistema Gestor de Bases de Datos o SGBD?” [Online]. Available: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>. [Accessed: 12-Jun-2018].
- [14] “¿Qué es MySQL? :: esepestudio, especialistas web.” [Online]. Available: <http://www.esepestudio.com/noticias/que-es-mysql>. [Accessed: 12-Jun-2018].
- [15] “Visual Paradigm para UML.” [Online]. Available: <file:///D:/Documents/Paginas%20utilizadas/Visual%20Paradigm%20para%20UML.htm>. [Accessed: 05-Jun-2017].
- [16] “Definición de html - Qué es, Significado y Concepto.” [Online]. Available: <https://definicion.de/html/>. [Accessed: 12-Jun-2018].
- [17] “¿Qué es CSS?” [Online]. Available: <http://www.maestrosdelweb.com/introcss/>. [Accessed: 12-Jun-2018].
- [18] V. Figueroa, “Características de PHP,” *Scribd*. [Online]. Available: <https://es.scribd.com/doc/50288837/Caracteristicas-de-PHP>. [Accessed: 10-May-2016].
- [19] “IBM Knowledge Center - Servidores de aplicaciones.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_73/rzahg/rzahgebappserv.htm. [Accessed: 12-Jun-2018].
- [20] “Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve? | Blog ibrugor.” [Online]. Available: <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>. [Accessed: 12-Jun-2018].
- [21] “¿Qué es un ‘framework’? | jordisan.net.” [Online]. Available: <https://jordisan.net/blog/2006/que-es-un-framework/>. [Accessed: 12-Jun-2018].
- [22] “¿Qué es Symfony?” [Online]. Available: <http://symfony.es/pagina/que-es-symfony/>. [Accessed: 12-Jun-2018].
- [23] “Qué es Bootstrap y cuáles son sus ventajas | Punto Abierto.” [Online]. Available: <https://puntoabierto.net/blog/que-es-bootstrap-y-cuales-son-sus-ventajas>. [Accessed: 12-Jun-2018].
- [24] M. Uvaldi, “Qué es JQuery y cómo implementarlo.” [Online]. Available: <http://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>. [Accessed: 12-Jun-2018].
- [25] “NetBeans.” [Online]. Available: <https://www.fdi.ucm.es/profesor/luis/fp/devtools/NetBeans.html>. [Accessed: 12-Jun-2018].
- [26] Universidad Tecnológica, “diagrama de casos de uso del negocio y del sistema,” 18:48:44 UTC.
- [27] “IBM Knowledge Center - Modelos de objetos de negocio.” [Online]. Available:

Referencias Bibliográficas

- https://www.ibm.com/support/knowledgecenter/es/SS69YH_6.0.0/com.ibm.sps.s.adv.doc/cads/concepts/business_object_models.html. [Accessed: 12-Jun-2018].
- [28] W. Bosh, “Requerimientos funcionales y no funcionales.” [Online]. Available: <https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>. [Accessed: 12-Jun-2018].
- [29] “DISEÑO DE BASE DE DATOS,” *INFORMÁTICA APLICADA*, 10-Jan-2012.
- [30] “IBM Knowledge Center - Modelos lógicos de datos.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SS9UM9_9.1.2/com.ibm.datatools.logical.ui.doc/topics/clogmod.html. [Accessed: 12-Jun-2018].
- [31] C. Castro, “IBM Knowledge Center - Modelos físicos de datos.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SS9UM9_9.1.0/com.ibm.datatools.core.ui.doc/topics/cphysmod.html. [Accessed: 12-Jun-2018].
- [32] “Artefacto: Modelo de despliegue.” [Online]. Available: https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/workproducts/rup_deployment_model_57DF1DF5.html. [Accessed: 12-Jun-2018].
- [33] B. Pérez Lamancha, “PRIS-GestionTesting-BPerez - CES-PRIS_Gestion de Pruebas Funcionales.pdf.” [Online]. Available: file:///E:/Documents/Tesis/Basilio/Documents/CES-PRIS_Gestion%20de%20Pruebas%20Funcionales.pdf. [Accessed: 10-May-2016].
- [34] “Pruebas funcionales y casos de prueba,” *Globe Testing*, 09-May-2016. .

Bibliografía

- [1] «Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve? | Blog ibrugor». [En línea]. Disponible en: <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>. [Accedido: 12-jun-2018].
- [2] «Arquitectura cliente/servidor». [En línea]. Disponible en: <http://isa.uniovi.es/domotica/Temas/T6/T6-ClienteServidor.htm>. [Accedido: 12-jun-2018].
- [3] «Artefacto: Modelo de despliegue». [En línea]. Disponible en: https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/workproducts/rup_deployment_model_57DF1DF5.html. [Accedido: 12-jun-2018].
- [4] «Cultivo - EcuRed». [En línea]. Disponible en: <https://www.ecured.cu/Cultivo>. [Accedido: 12-jun-2018].
- [5] «Definición de html - Qué es, Significado y Concepto». [En línea]. Disponible en: <https://definicion.de/html/>. [Accedido: 12-jun-2018].
- [6] Universidad Tecnológica, «diagrama de casos de uso del negocio y del sistema», 18:48:44 UTC.
- [7] «DISEÑO DE BASE DE DATOS», *INFORMÁTICA APLICADA*, 10-ene-2012. .
- [8] «El Proceso Unificado de Desarrollo de Software (RUP)». [En línea]. Disponible en: <http://yaqui.mx.l.uabc.mx/~molguin/as/RUP.htm>. [Accedido: 12-jun-2018].
- [9] «IBM Knowledge Center - Modelos de objetos de negocio». [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SS69YH_6.0.0/com.ibm.spss.adv.doc/cads/concepts/business_object_models.html. [Accedido: 12-jun-2018].
- [10] «IBM Knowledge Center - Modelos físicos de datos». [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SS9UM9_9.1.0/com.ibm.datatools.core.ui.doc/topics/cphysmod.html. [Accedido: 12-jun-2018].
- [11] «IBM Knowledge Center - Modelos lógicos de datos». [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SS9UM9_9.1.2/com.ibm.datatools.logical.ui.doc/topics/clogmod.html. [Accedido: 12-jun-2018].
- [12] «IBM Knowledge Center - Patrón de diseño de modelo-vista-controlador». [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm. [Accedido: 12-jun-2018].
- [13] «IBM Knowledge Center - Servidores de aplicaciones». [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_73/rzahg/rzahgebappserv.htm. [Accedido: 12-jun-2018].
- [14] «Metodologías del Desarrollo de Software». [En línea]. Disponible en:

- <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>. [Accedido: 12-jun-2018].
- [15] «Modelo de Casos de Uso del Sistema». [En línea]. Disponible en: <http://www.eumed.net/libros-gratis/2009c/585/Modelo%20de%20Casos%20de%20Uso%20del%20Sistema.htm>. [Accedido: 12-jun-2018].
- [16] «Modelos de despliegue cloud: Cloud privado, cloud público y cloud híbrido | Nexica». [En línea]. Disponible en: <https://www.nexica.com/es/blog/modelos-de-despliegue-cloud-cloud-privado-cloud-p%C3%ABablico-y-cloud-h%C3%ADbrido>. [Accedido: 12-jun-2018].
- [17] «NetBeans». [En línea]. Disponible en: <https://www.fdi.ucm.es/profesor/luis/fp/devtools/NetBeans.html>. [Accedido: 12-jun-2018].
- [18] «Plagas y enfermedades de las plantas : FAO in Emergencias». [En línea]. Disponible en: <http://www.fao.org/emergencias/tipos-de-peligros-y-de-emergencias/plagas-y-enfermedades-de-las-plantas/es/>. [Accedido: 12-jun-2018].
- [19] «Qué es Bootstrap y cuáles son sus ventajas | Punto Abierto». [En línea]. Disponible en: <https://puntoabierto.net/blog/que-es-bootstrap-y-cuales-son-sus-ventajas>. [Accedido: 12-jun-2018].
- [20] «¿Qué es CSS?» [En línea]. Disponible en: <http://www.maestrosdelweb.com/introcss/>. [Accedido: 12-jun-2018].
- [21] «Qué es el lenguaje unificado de modelado (UML) | Lucidchart». [En línea]. Disponible en: <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml>. [Accedido: 12-jun-2018].
- [22] «Qué es JQuery y cómo implementarlo». [En línea]. Disponible en: <http://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>. [Accedido: 12-jun-2018].
- [23] «¿Qué es MySQL? :: esepestudio, especialistas web». [En línea]. Disponible en: <http://www.esepestudio.com/noticias/que-es-mysql>. [Accedido: 12-jun-2018].
- [24] «Qué es PHP». [En línea]. Disponible en: <https://www.desarrolloweb.com/articulos/392.php>. [Accedido: 12-jun-2018].
- [25] «¿Qué es Symfony?» [En línea]. Disponible en: <http://symfony.es/pagina/que-es-symfony/>. [Accedido: 12-jun-2018].
- [26] «Qué es UML? | LENGUAJE DE MODELADO UNIFICADO UML». [En línea]. Disponible en: http://stadium.unad.edu.co/ovas/10596_9839/qu_es_uml.html. [Accedido: 12-jun-2018].
- [27] «¿Qué es un “framework”? | jordisan.net». [En línea]. Disponible en: <https://jordisan.net/blog/2006/que-es-un-framework/>. [Accedido: 12-jun-2018].
- [28] «¿Qué es un Sistema Gestor de Bases de Datos o SGBD?» [En línea]. Disponible en:

- <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>. [Accedido: 12-jun-2018].
- [29] «Qué es y para qué sirve el lenguaje CSS (Cascading Style Sheets - Hojas de Estilo).» [En línea]. Disponible en: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&catid=46&Itemid=163. [Accedido: 12-jun-2018].
- [30] «¿Qué son las TIC?» [En línea]. Disponible en: <http://tutorial.cch.unam.mx/bloque4/lasTIC>. [Accedido: 12-jun-2018].
- [31] «Requerimientos funcionales: Ejemplos - La Oficina de Proyectos de Informática». [En línea]. Disponible en: <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>. [Accedido: 12-jun-2018].
- [32] «Requerimientos funcionales y no funcionales». [En línea]. Disponible en: <https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>. [Accedido: 12-jun-2018].
- [33] «SAVIA. Un sistema web de ayuda a la... (PDF Download Available)». [En línea]. Disponible en: https://www.researchgate.net/publication/295519751_SAVIA_Un_sistema_web_de_ayuda_a_la_decision_para_control_de_plagas. [Accedido: 12-jun-2018].
- [34] «Tips de Desarrollo Web | Arquitectura N-Capas y LinqToSQL». [En línea]. Disponible en: <http://www.nicholls.co/blog/post/Arquitectura-N-Capas-y-LinqToSQL>. [Accedido: 12-jun-2018].
- [35] «Una “app” para los agricultores y ganaderos», *Agronews Castilla y León*. [En línea]. Disponible en: <http://www.agronewscastillayleon.com/una-app-para-los-agricultores-y-ganaderos>. [Accedido: 12-jun-2018].
- [36] «Una “app” para los agricultores y ganaderos | Agronews Castilla y León». [En línea]. Disponible en: <http://www.agronewscastillayleon.com/una-app-para-los-agricultores-y-ganaderos>. [Accedido: 12-jun-2018].
- [37] «Vigilancia Fitosanitaria». [En línea]. Disponible en: <https://www.senasa.gob.pe/senasa/vigilancia-fitosanitaria/>. [Accedido: 12-jun-2018].
- [38] «Vigilancia Fitosanitaria». [En línea]. Disponible en: <http://www.cnmsf.gob.do/Generalidades/An%C3%A1lisisdeRiesgoyVigilanciaEpidemi%C3%B3logica/VigilanciaFitosanitaria/tabid/157/Default.aspx>. [Accedido: 12-jun-2018].
- [39] «Visual Paradigm For Uml». [En línea]. Disponible en: <https://es.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>. [Accedido: 12-jun-2018].

Glosario de términos.

BD (Base de Datos): Banco de información que contiene diversos datos categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

CU (Caso de Uso): Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Framework: Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software.

IDE (Entorno de Desarrollo Integrado): Aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

PDF (Formato de Documento Portátil): Formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

SQL (Lenguaje de Consulta Estructurado): Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

TCP/IP: El conjunto TCP/IP tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores web.

XML (Lenguaje de Marcas Extensible): Utilizado para almacenar datos en forma legible. Permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

MINAG: Ministerio de la Agricultura.

Glosario de términos

EPP: Estación de Protección de Plantas.

*Anexos***Anexo A: Descripción de los Casos de Uso del Sistema.****A.1 Descripción textual del caso de uso Autenticarse.**

Nombre del Caso de Uso	Autenticarse
Actores	Trabajador
Resumen	El caso de uso se inicia en el momento que el usuario introduce sus datos (Usuario y Contraseña), el Sistema verifica si los datos introducidos son correctos y procede a validar si existen en la BD, de ser así el usuario accede al sistema con un determinado rol. De no existir el usuario o los datos introducidos son incorrectos, el Sistema muestra un mensaje de error. Siempre que ocurra un error, el sistema redirecciona al usuario a la página de autenticación. El caso de uso termina cuando el usuario se autentica.
Requisitos Asociados	R1
Precondiciones	El usuario debe estar registrado en la BD del Sistema.
Postcondiciones	Después de acceder al sistema el usuario solo puede realizar las acciones correspondientes a su rol.

A.2 Descripción textual del Caso de Cerrar Sesión.

Nombre del Caso de Uso	Cerrar Sesión
Actores	Trabajador
Resumen	El caso de uso se inicia en el momento que el usuario desea salir de la sesión. El caso de uso termina cuando el usuario se autentica.
Requisitos Asociados	R2
Precondiciones	El usuario debe estar logueado en el sistema.
Postcondiciones	La sesión quedará cerrada para el usuario logueado.

A.3 Descripción textual del Caso de Uso Cambiar Contraseña.

Nombre del Caso de Uso	Cambiar Contraseña
Actores	Trabajador
Resumen	El caso de uso se inicia en el momento que el usuario toma la decisión de cambiar su contraseña. Para ello debe insertar su nueva contraseña y la confirmación de la misma. El caso de uso termina cuando los datos hayan sido guardados.

Requisitos Asociados	R3
Precondiciones	El usuario debe estar logueado en el sistema.
Postcondiciones	La sesión quedará cerrada para el usuario logueado.

A.4 Descripción textual del Caso de Uso Gestionar Usuario.

Nombre del Caso de Uso	Gestionar Usuario
Actores	Administrador
Resumen	El caso de uso se inicia en el actor Administrador accede al botón de gestión de usuarios. Acto seguido el Administrador podrá listar, editar, eliminar o buscar un usuario del sistema. A partir de la opción que el Administrador seleccione el sistema mostrará un mensaje de error o de éxito dependiendo de la acción escogida de la manera en que esta sea realizada. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R4, R5, R6, R7, R8
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará creado, actualizado o eliminado un Usuario en el sistema por el actor Administrador.

A.5 Descripción textual del Caso de Uso Gestionar Municipio.

Nombre del Caso de Uso	Gestionar Municipio
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear, editar o eliminar un municipio. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R9, R10, R11, R12, R13
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará creado, actualizado o eliminado un Municipio en el sistema por el Profesional.

A.6 Descripción textual del Caso de Uso Gestionar Empresa.

Nombre del Caso de Uso	Gestionar Empresa
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear,

	editar o eliminar una Empresa. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R14, R15, R16, R17, R18
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedarácreada, actualizada o eliminada una Empresa en el sistema por el Profesional.

A.7 Descripción textual del Caso de Uso Gestionar Rama.

Nombre del Caso de Uso	Gestionar Rama
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear, editar o eliminar una Rama. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R14, R15, R16, R17, R18
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará creada, actualizada o eliminada una Rama en el sistema por el Profesional.

A.8 Descripción textual del Caso de Uso Gestionar Unidad.

Nombre del Caso de Uso	Gestionar Unidad
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear, editar o eliminar una Unidad. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R14, R15, R16, R17, R18
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedarácrear, actualizada o eliminada una Unidad en el sistema por el Profesional.

A.9 Descripción textual del Caso de Uso Gestionar Estación.

Nombre del Caso de Uso	Gestionar Estación
-------------------------------	--------------------

Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear, editar o eliminar una Estación. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R28, R29, R30, R31, R32
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará crear, actualizada o eliminada una Estación en el sistema por el Profesional.

A.10 Descripción textual del Caso de Uso Gestionar Cultivo.

Nombre del Caso de Uso	Gestionar Cultivo
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear, editar o eliminar un Cultivo. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R37, R38, R39, R40
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará creado, actualizado o eliminado un Cultivo en el sistema por el Profesional.

A.11 Descripción textual del Caso de Uso Gestionar Agente Nocivo.

Nombre del Caso de Uso	Gestionar Agente Nocivo
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesitacrear, editar o eliminar unAgente Nocivo. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R40, R41, R42, R43, R44
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará creado, actualizado o eliminado unAgente Nocivo en el sistema por el Profesional.

A.12 Descripción textual del Caso de Uso Gestionar Modelo.

Nombre del Caso de Uso	Gestionar Modelo
Actores	Profesional
Resumen	El caso de uso se inicia en el Profesional necesita crear, editar o eliminar un nuevo modelo. El sistema mostrará un mensaje éxito o error en el caso de la información esté correcta, o incorrecta o haya sido omitido algún campo. El caso de uso termina cuando los datos hayan sido actualizados.
Requisitos Asociados	R55, R56, R57, R58, R59
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	Quedará crear, actualizado o eliminado un Modelo en el sistema por el Profesional.

A.13 Descripción textual del Caso de Uso Visualizar Modelo 2004.

Nombre del Caso de Uso	Reportar Modelo 2004
Actores	Profesional
Resumen	El caso de uso se inicia en el usuario necesita realizar un análisis de la información recogida en los diferentes modelos, exportar a formato PDF. El sistema mostrará un mensaje de éxito cuando visualice y exporte los modelos deseados, o un mensaje de error en el caso de que no existan modelos creados. El caso de uso termina cuando el usuario exportó el modelo deseado.
Requisitos Asociados	R54, R55, R56
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	El usuario visualizó y exportó el modelo 2004.

A.14 Descripción textual del Caso de Uso Visualizar Modelo 2004d.

Nombre del Caso de Uso	Reportar Modelo 2004d
Actores	Profesional
Resumen	El caso de uso se inicia en el usuario necesita realizar un análisis de la información recogida en los diferentes modelos, exportar a formato PDF. El sistema mostrará un mensaje de éxito cuando visualice y exporte los modelos deseados, o un mensaje de error en el caso de que no existan modelos creados. El caso de uso termina cuando el usuario exportó el modelo deseado.
Requisitos Asociados	R57, R58, R59
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	El usuario visualizó y exportó el modelo 2004d.

A.15 Descripción textual del Caso de Uso Visualizar Modelo 2003.

Nombre del Caso de Uso	Reportar Modelo 2003
Actores	Profesional
Resumen	El caso de uso se inicia en el usuario necesita realizar un análisis de la información recogida en los diferentes modelos, exportar a formato PDF. El sistema mostrará un mensaje de éxito cuando visualice y exporte los modelos deseados. El caso de uso termina cuando el usuario exportó el modelo deseado.
Requisitos Asociados	R60, R61, R62
Precondiciones	El usuario debe estar logueado en el Sistema.
Postcondiciones	El usuario visualizó y exportó el modelo 2003.

Anexo B: Prototipos de Interfaz.

Anexo P1: Autenticarse.

El prototipo de interfaz de usuario para el login de GESAVE se muestra en un recuadro gris. En la parte superior, el título "GESAVE" está escrito en grandes letras negras. Debajo del título, hay un campo de texto con el texto "Ingresar" en gris. A continuación, hay dos campos de entrada de texto: el primero está etiquetado "Usuario" y el segundo "Contraseña". Debajo de estos campos, hay un botón rectangular verde con el texto "Acceder" en blanco.

Anexo P2: Cambiar Contraseña.

Contraseña

Repetir Contraseña

Anexo P3: Gestionar Usuario.

Crear usuario

Usuario	Nombre
<input type="text" value="Nombre de usuario"/>	<input type="text" value="Ingrese su nombre"/>
Apellidos	Correo electrónico
<input type="text" value="Ingrese sus apellidos"/>	<input type="text" value="Ingrese su correo electrónico"/>
Contraseña	Repetir Contraseña
<input type="text" value="Ingrese una contraseña"/>	<input type="text" value="Ingrese una contraseña"/>
Rol	
<input type="text" value="Seleccione un rol"/>	
<input checked="" type="checkbox"/> Activo	
<input type="button" value="Guardar"/>	

Anexo P4: Gestionar Municipio.

Crear Municipio

Nombre municipio

Provincia

Anexo P5: Gestionar Empresa.

Crear empresa

Nombre empresa

Rama

Municipio

Anexo P6: Gestionar Rama.

Crear Rama

Nombre rama

Anexo P7: Gestionar Unidad.

Crear unidad

Nombre unidad	Empresa
<input type="text"/>	<input type="text" value="Seleccione la empresa"/>

Anexo P8: Gestionar Epp.

Crear Epp

Nombre EPP	Dirección
<input type="text"/>	<input type="text"/>
Municipio	Provincia
<input type="text" value="Seleccione el municipio"/>	<input type="text" value="Seleccionar provincia"/>

Anexo P9: Gestionar Cultivo.

Crear cultivo

Nombre cultivo

Agentes nocivos

[Guardar](#)

[Regresar a la lista](#)

Anexo P10: Gestionar Agente Nocivo.

Crear Plaga

Nombre

Grupo

Cultivos

[Guardar](#)

[Regresar a la lista](#)

Anexo P11: Gestionar Producto.

Crear producto

Nombre

Tipo producto

[Guardar](#)

[Regresar a la lista](#)

Anexo P12: Gestionar Modelo 2004.

Crear Modelo 2004

Categoría <input type="text"/>	Fecha <input type="text"/>	Unidad <input type="text" value="Seleccionar unidad"/>
Epp <input type="text" value="Seleccionar EPP"/>	Areas Existentes <input type="text"/>	Areas Observadas <input type="text"/>
Areas señalizadas <input type="text"/>	Areas bajo pronóstico <input type="text"/>	Area protegida <input type="text"/>

Anexo P13: Gestionar Modelo 2004d.

Crear modelo M2004d

Fecha <input type="text"/>	Cultivo <input type="text" value="Selecciona un cultivo"/>	EPP <input type="text" value="Selecciona un EPP"/>
Categoría <input type="text"/>	Existentes <input type="text"/>	Observado <input type="text"/>
Decena <input type="text"/>		

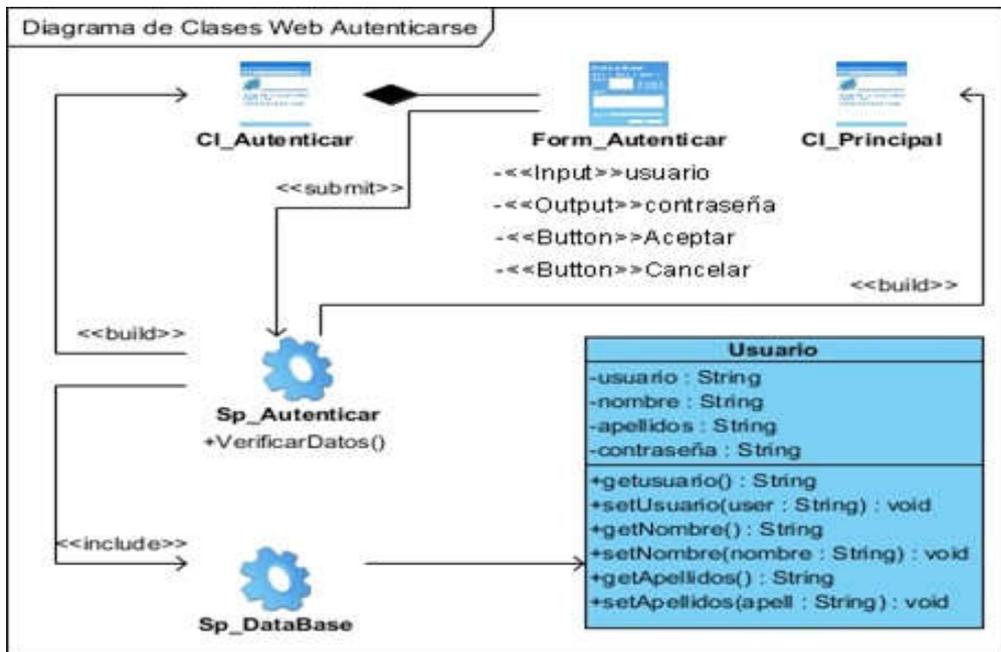
Anexo P14: Gestionar Modelo 2003.

Crear Modelo 2003

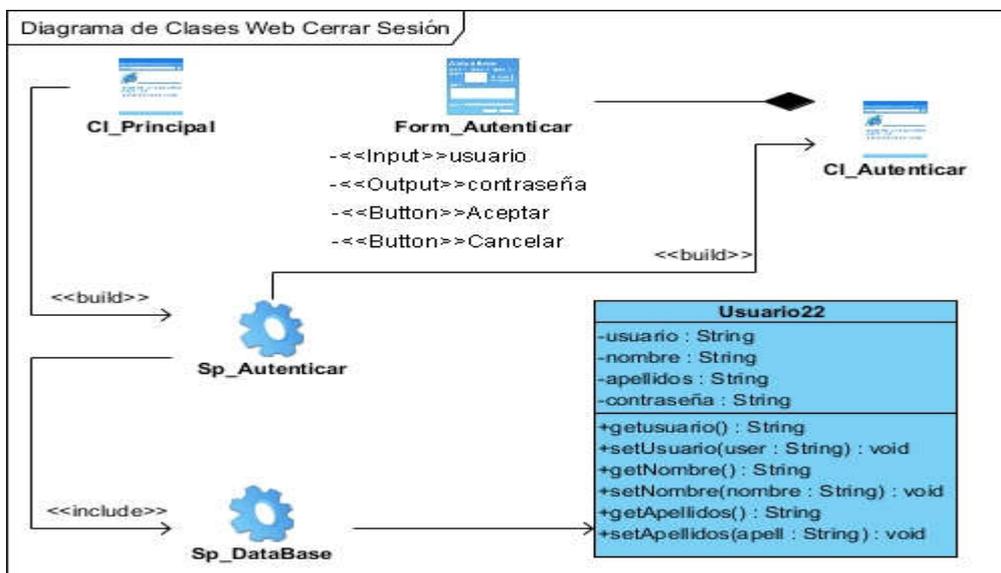
Fecha <input type="text"/>	Categoría <input type="text"/>	Nombre unidad <input type="text" value="Seleccionar unidad"/>
EPP <input type="text" value="Selecciona un EPP"/>	Nombre unidad <input type="text"/>	Plan <input type="text"/>
Zonas visitadas <input type="text"/>	Reinspeccionada <input type="text"/>	Trimestre <input type="text" value="Seleccione un Trimestre"/>

Anexo C: Diagramas de Clase Web.

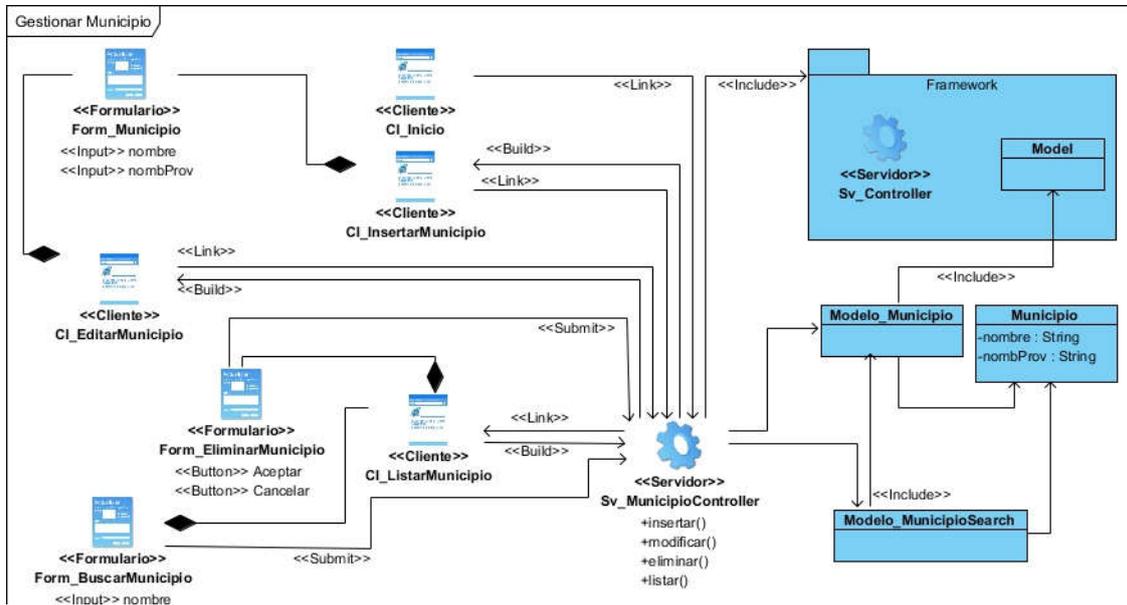
Anexo C1: Autenticarse.



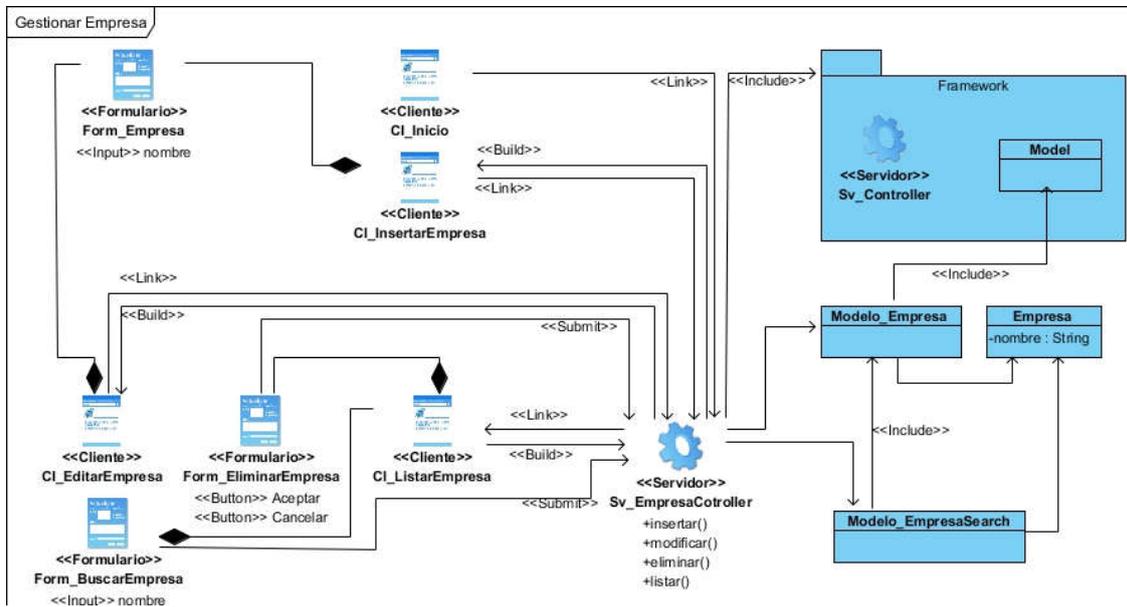
Anexo C2: Cerrar sesión.



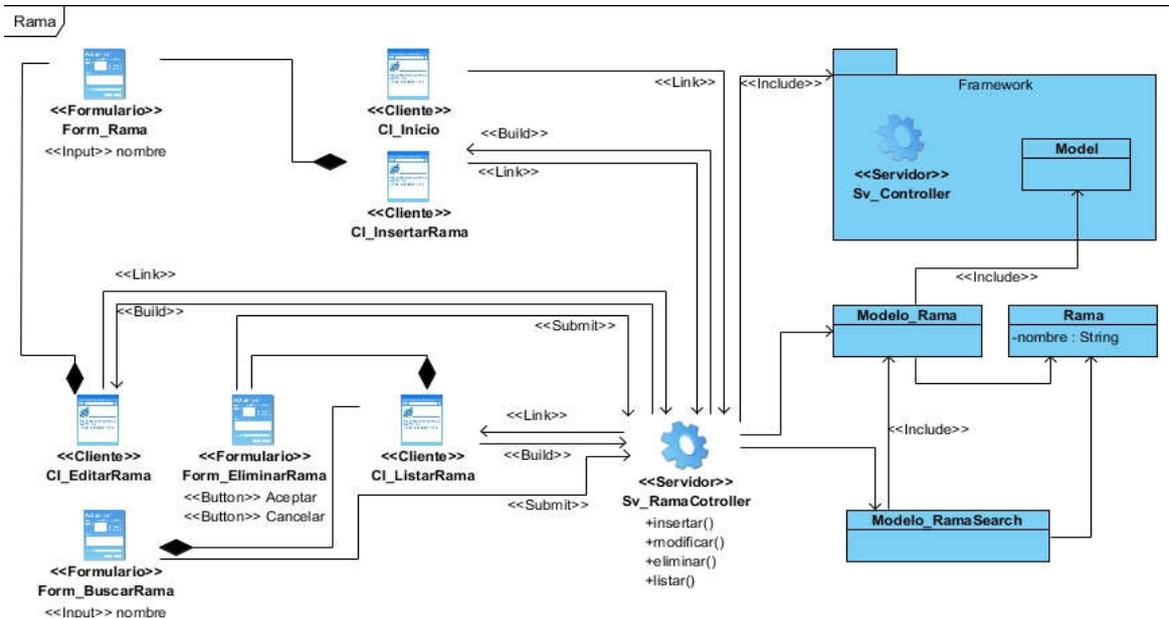
Anexo C5: Gestionar Municipio.



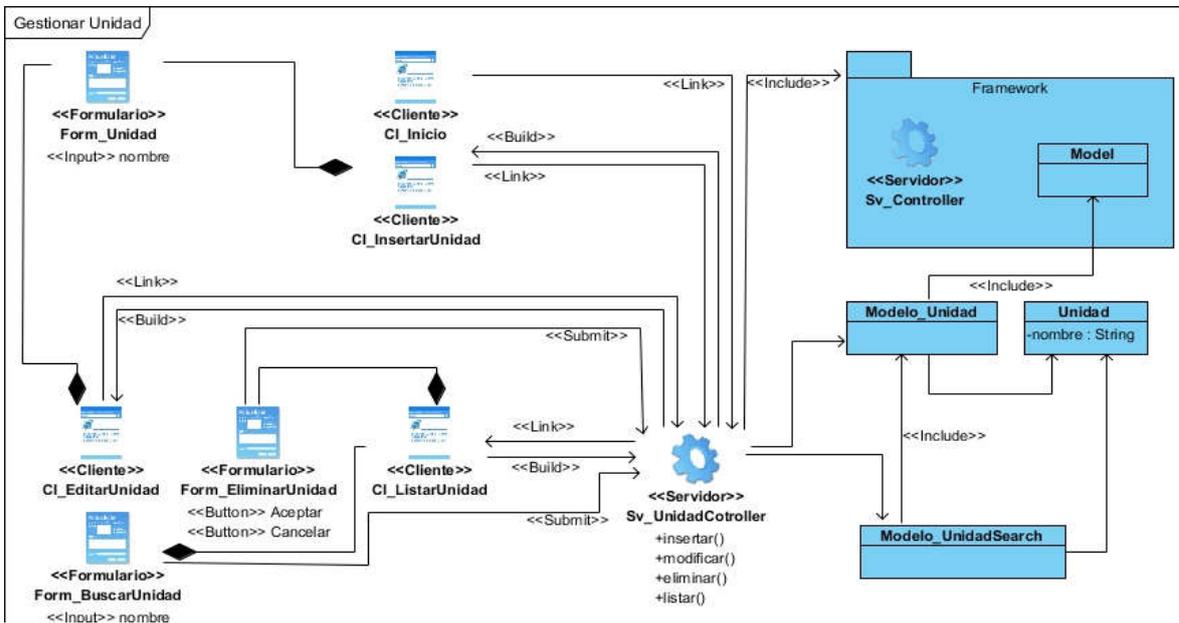
Anexo C6: Gestionar Empresa.



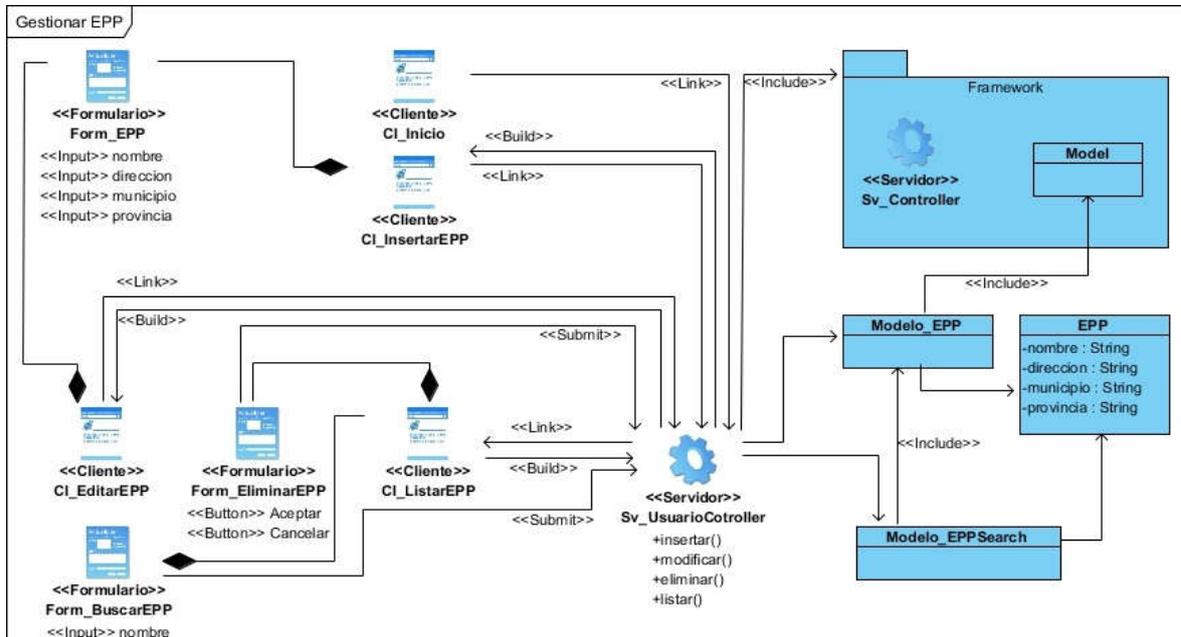
Anexo C7: Gestionar Rama.



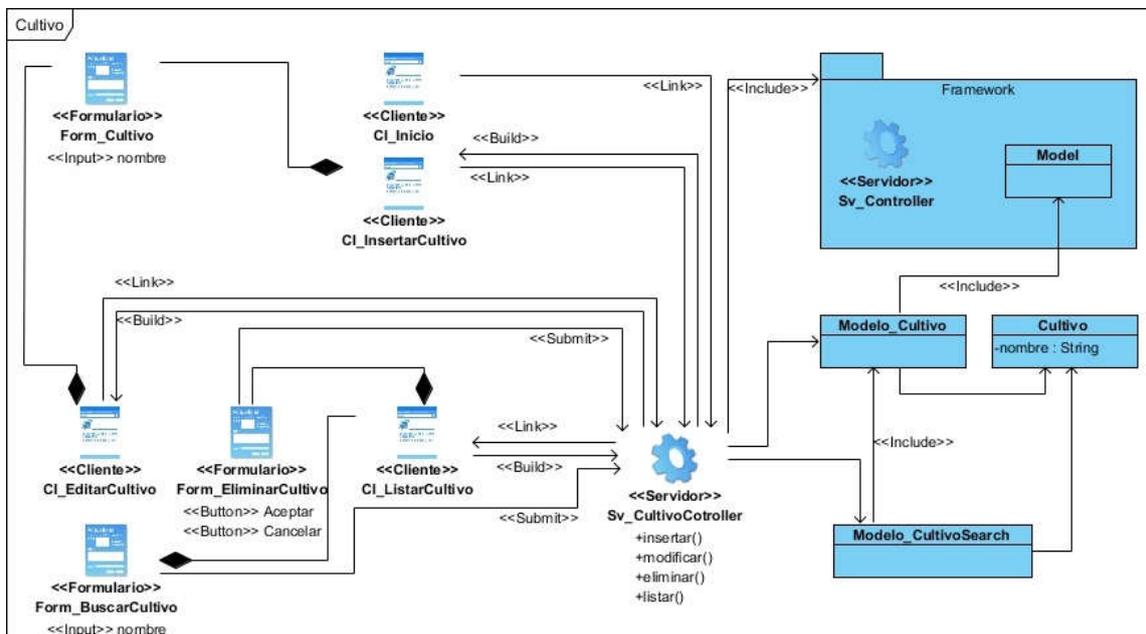
Anexo C8: Gestionar Unidad.



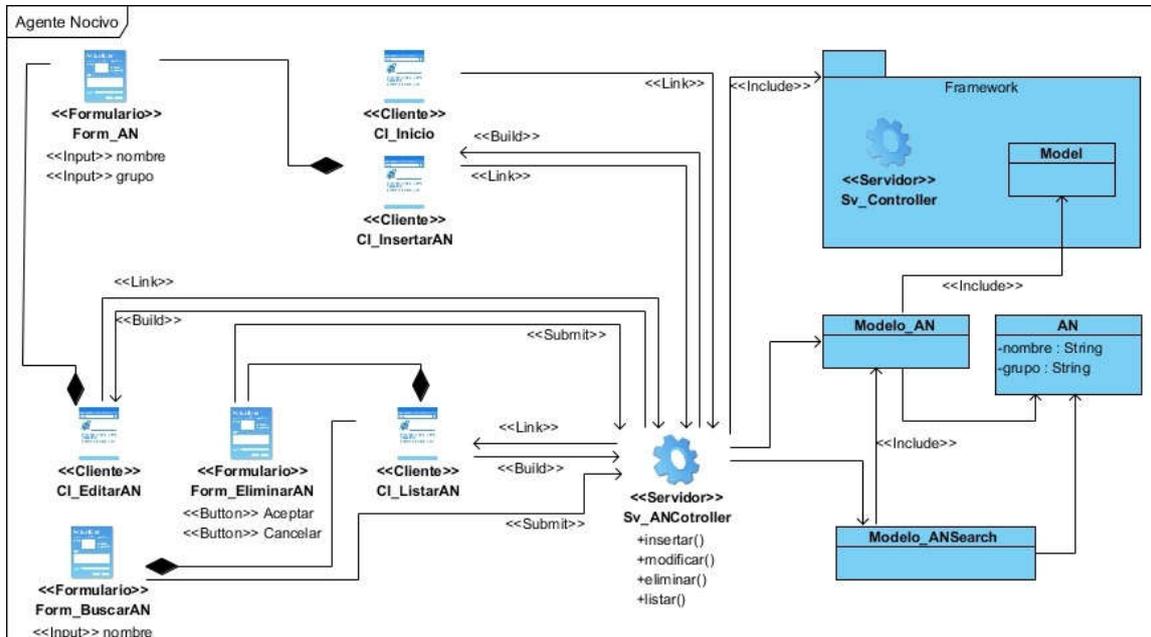
Anexo C9: Gestionar EPP.



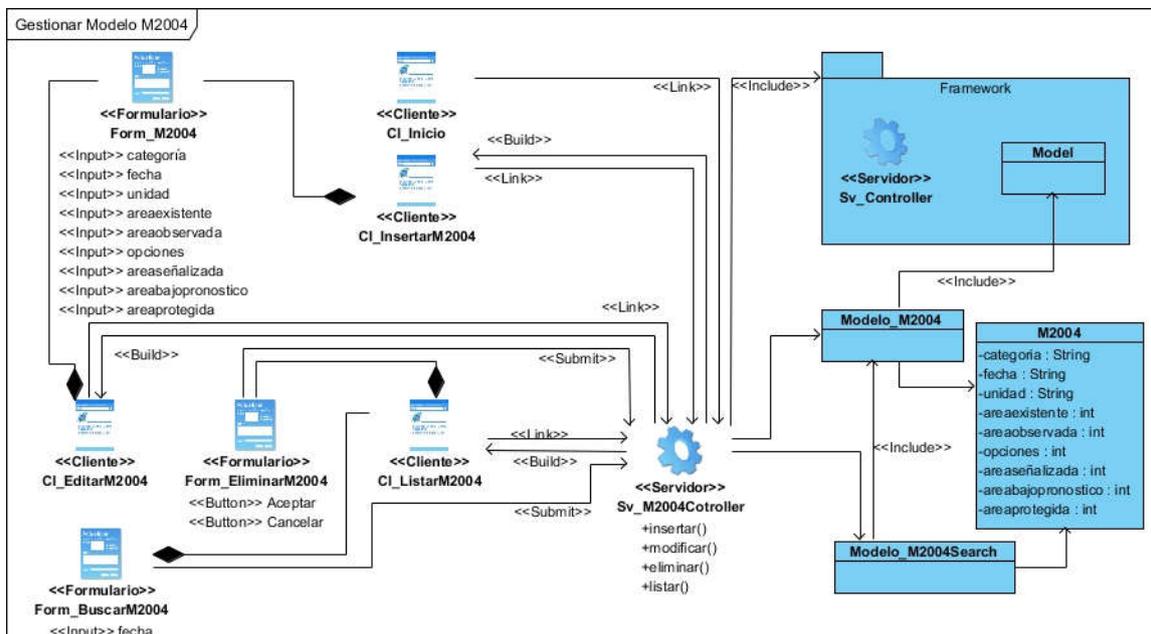
Anexo C10: Gestionar Cultivo.



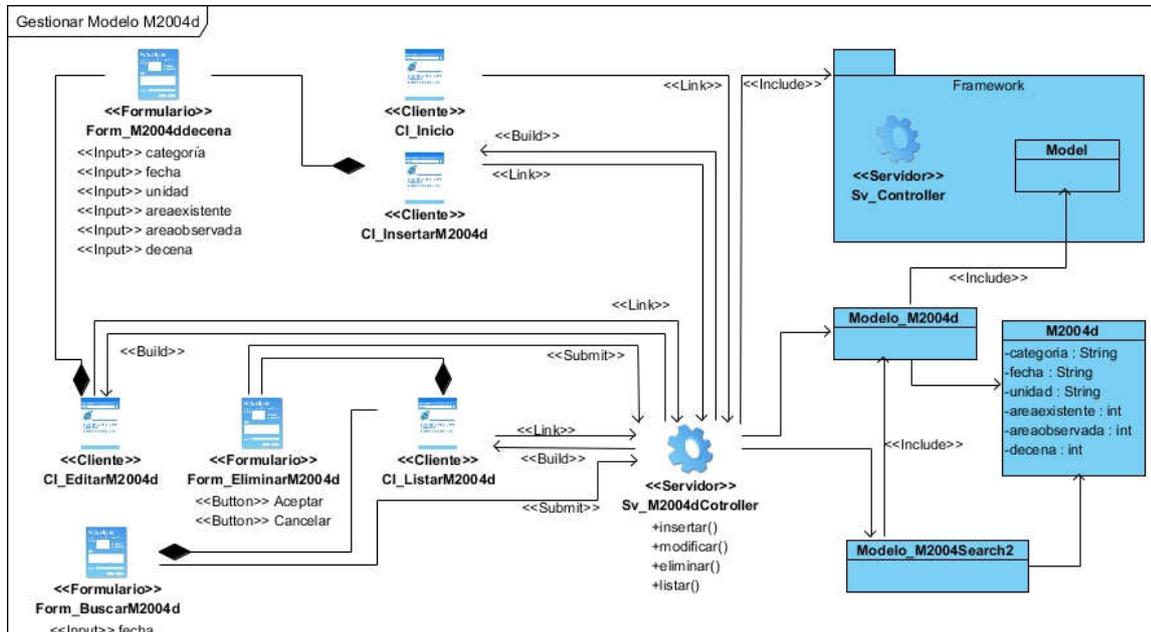
Anexo C11: Agente Nocivo.



Anexo C12: Gestionar Modelo 2004.



Anexo C13: Gestionar Modelo 2004d.



Anexo C14: Gestionar Modelo 2003.

