

Universidad de las Ciencias Informáticas
Facultad de Ciencias y Tecnologías Computacionales



Título: Sistema de información geográfica para el monitoreo y control de recursos agrícolas (SIGMCRA)

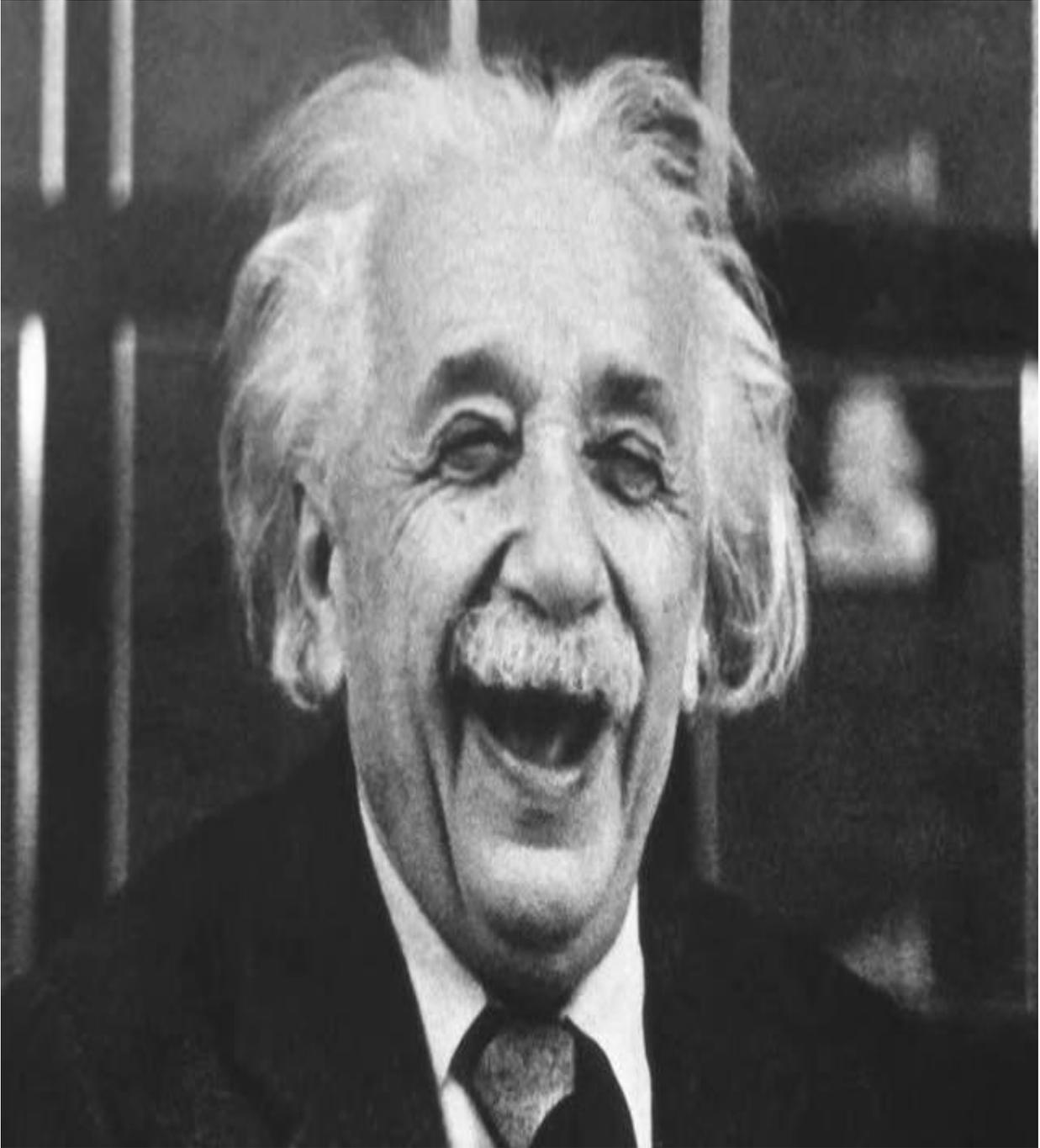
Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas:

Autores: Enmanuel Adalberto Naranjo Nápoles

José Rubén Machado Vega

Tutor: Ing. Ernesto Camilo Gonzáles Casanova

La Habana, junio de 2017



Si quieres describir la verdad hazlo con sencillez, y la elegancia déjasela al sastre.

Albert Einstein.

DECLARACIÓN DE AUTORIA

Nosotros declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que se así conste firmamos la presente a los _____ días del mes de _____ del año_____.

Enmanuel A. Naranjo Nápoles

Firma del Autor

José Rubén Machado Vega

Firma del Autor

MsC. Yuniel Eliades Proenza Arias

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ernesto Camilo González Casanova

Facultad: Facultad de Ciencias y Tecnologías Computacionales

Correo electrónico: ernestocamilo@uci.cu

Teléfono: 835-874

Dedicatoria

Enmanuel A. Naranjo Napoles.

A mi madre, por hacerme quien soy, por amarme tanto y hacerme sentir orgulloso, de mí, de ella y de toda la familia. A mis abuelos, por estar ahí desde que tengo memoria apoyándome y educándome para ser el hombre que hoy soy, a mi abuelo por regañarme siempre cuando hacía algo malo y no malcriarme como hacen algunos abuelos, a mi abuela por todo su amor y su cuidado, por tantas charlas en las que aprendimos mutuamente y muy especialmente por estar ahí siempre para mi madre. A mi hermana Rosalba, por ser siempre aquel bastión donde ahogar mis penas y compartir mis alegrías, más que mi hermana, es mi segunda madre. A mi hermana Yinet por cuidar siempre a mi padre en mi ausencia y por quererme tanto. A mi tío Juan, que me enseñó lo resabiosos que son los Napoles pero que fue durante mucho tiempo mi modelo a seguir, que me enseñó de responsabilidad y hogar, de lo difícil que es la vida de adulto y lo importante de un gran corazón. A mi tío Berto por ser mi padrino y siempre defenderme de mi abuelo aun cuando hacia maldades de chiquillo, también por toda la ayuda que le ha dado a mi hermana, a mi mama, a mis abuelos y a mí. A mi padre por quererme y regresar a mi después de tanto tiempo. A mi tía Dulce por estar ahí para mí, por preocuparse por mí y por sentirse orgullosa. También para la única persona que no se encuentra físicamente entre nosotros, a mi abuelo Mongo, el único que se encuentra fuera del alcance de estas líneas, y para quien las escribo aún más fuertes, lo suficiente para que te lleguen a donde estés. A mis primos y primas, y a toda la familia, a Nere, Maida, Mirna, a mi novia Lilibeth, a mi único amigo verdadero, Emilio, a mi profe Norma y a mis padrastros, a todos los que influyeron en mi educación y enseñanza, a todos ustedes les dedico esta tesis.

José Rubén Machado Vega.

A mis padres por ser el motor que me impulsa a ser cada día mejor.

Agradecimientos

Enmanuel A. Naranjo Napoles

A mi madre, por hacerme quien soy, por amarme tanto y hacerme sentir orgulloso, de mí, de ella y de toda la familia. A mis abuelos, por estar ahí desde que tengo memoria apoyándome y educándome para ser el hombre que hoy soy, a mi abuelo por regañarme siempre cuando hacía algo malo y no malcriarme como hacen algunos abuelos, a mi abuela por todo su amor y su cuidado, por tantas charlas en las que aprendimos mutuamente y muy especialmente por estar ahí siempre para mi madre. A mi hermana Rosalba, por ser siempre aquel bastión donde ahogar mis penas y compartir mis alegrías, más que mi hermana, es mi segunda madre. A mi hermana Yinet por cuidar siempre a mi padre en mi ausencia y por quererme tanto. A mi tío Juan, que me enseñó lo resabiosos que son los Napoles pero que fue durante mucho tiempo mi modelo a seguir, que me enseñó de responsabilidad y hogar, de lo difícil que es la vida de adulto y lo importante de un gran corazón. A mi tío Berto por ser mi padrino y siempre defenderme de mi abuelo aun cuando hacia maldades de chiquillo, también por toda la ayuda que le ha dado a mi hermana, a mi mama, a mis abuelos y a mí. A mi padre por quererme y regresar a mi después de tanto tiempo. A mi tía Dulce por estar ahí para mí, por preocuparse por mí y por sentirse orgullosa. También para la única persona que no se encuentra físicamente entre nosotros, a mi abuelo Mongo, el único que se encuentra fuera del alcance de estas líneas, y para quien las escribo aún más fuertes, lo suficiente para que te lleguen a donde estés. A mis primos y primas, y a toda la familia, a Nere, Maida, Mirna, a mi novia Lilibeth por estar a mi lado y apoyarme constantemente, a mi único verdadero amigo, Emílio, a mis padrastros, a mi profe Norma y a todos los que influyeron en mi educación y enseñanza. A mis vecinos por ser los mejores de todos los vecinos, a mis compañeros de clase de esta y otras enseñanzas ya que todos han dejado alguna huella en mí. Quiero agradecer también, y lo hago como tributo, a Marti, a Fidel, al Che y a todos y cada uno de los que pagaron con su sangre, no solo el precio de la escuela, la salud y la Revolución, sino también las risas y lágrimas que puedan emerger aquí. Agradezco a los presentes, al tribunal, a la oponente, a mi tutor por su esfuerzo y dedicación, todos me ayudaron con sus observaciones y recomendaciones. Especialmente, quiero añadir, agradezco a mi compañero de tesis por su apoyo, confianza y dedicación. Como amigo y compañero. La lista es larga y extensa, no todos han de nombrarse, pero he de terminar. A los que faltan, no les agradeceré en palabras pues no alcanza el papel, pero pueden estar seguros que tengo espacio para ustedes en el corazón.

José Rubén Machado Vega

Quiero agradecer a mis padres por ser el ejemplo que me guía por estar presente cada vez que los necesite y por confiar en mi quiero que sepan que mis logros son también sus logros, a Cristina por ser la madrastra que todo el mundo desearía tener, gracias por ayudarme siempre, a mi tutor por todas las ayudas y explicaciones dadas durante el desarrollo de la aplicación, a mi compañero de tesis por confiar en mi a la hora de realizar esta tarea que nos definirá como ingenieros, a todos aquellos que me supieron explicar cuando tenía alguna duda, a todos los profesores que tuve durante toda la carrera que me supieron

inculcar todos los conocimientos necesarios para llegar aquí hoy, a todos los que de una forma u otra aportaron su granito de arena en el desarrollo de esta tesis y a todos los presentes muchas gracias por asistir.

Resumen

La falta de planificación que se presenta en el sector agropecuario cubano ha ocasionado un deterioro paulatino en las producciones agrícolas. Las transformaciones que se han realizado en la agricultura en los últimos años en Cuba han sido innumerables, sin embargo, los resultados pudieran ser mejores en términos productivos. Actualmente, en este sector, se carecen de mecanismos que permitan construir, rápida y eficientemente, diagnósticos sobre cada área de producción. Estos diagnósticos, aunque sí se realizan, generalmente demoran mucho tiempo en terminarse, por lo que en ocasiones es muy tarde para prevenir los daños. Con el fin de dar solución a estos problemas se decide realizar una investigación dirigida principalmente al desarrollo de una solución informática que permita aumentar la capacidad de monitoreo y control que se tienen sobre los procesos agrícolas e insumos de cada una de las entidades agropecuarias del país. Mediante la aplicación de los métodos analítico-sintético, análisis documental y la entrevista se determinan los principales elementos a documentar, así como las pautas que rigen la investigación. Se selecciona también una metodología de desarrollo y se dispone un equipo de trabajo. Como resultado se tiene un sistema informático que permite monitorear y aumentar el control que se tiene sobre los procesos agrícolas del país, contribuyendo así al aumento del rendimiento agrícola de las entidades agropecuarias de la nación y la documentación asociada a este.

Palabras clave: Sistema de información geográfica, procesos, planificación, control, agricultura de precisión.

Abstract

The lack of planning present in the Cuban agricultural sector has caused a gradual deterioration in the agricultural production. The transformations made in agriculture in the late years in Cuba have been innumerable, however, results could be better in productive terms. At present, in this sector, there is a lack of mechanisms to carry out, fast and efficiently, diagnoses on each area of production. These diagnostics, although done, generally take too much time to complete, being it that in occasions it is too late to prevent the damage. With the aim of giving solution to these problems has been decided to initiate an investigation guided principally to the development of an information-technology system which increases the monitoring and control capability over the agricultural processes and resources of the agricultural entities of the country. Applying the analytical synthetic, documentary analysis and the interview methods have been determined the main elements to document, as well as the guidelines that rule the investigation. A methodology of development is also selected and a work group is disposed. As a result, there is a computer system that allows monitoring and increases control over the country's agricultural processes, thus contributing to the increase in the agricultural performance of the nation's agricultural entities and the associated documentation.

Keywords: Geographical information system, processes, planning, control, precision agriculture.

Índice:

Introducción:	1
Métodos de investigación	4
Métodos teóricos	5
Métodos empíricos.....	5
Capítulo 1: Marco y base teórica.....	7
1.1 Introducción	7
1.2 Conceptos asociados al Problema	7
1.3 Elementos básicos de un SIG	10
1.4 Representar Datos en un SIG.....	10
1.4.1 Ventajas del Modelo Raster	11
1.4.2 Desventajas del Modelo Raster	11
1.4.3 Ventajas del Modelo Vectorial	12
1.4.4 Desventajas del Modelo Vectorial.....	12
1.3.4 Soluciones Existentes	12
1.5.1 GeoAgro GIS.....	12
1.5.2 PixAgri.....	13
1.5.3 Sistema de información geográfica para la gestión de cultivos en Cuba	14
1.5.4 Sistema de información geográfica para la agricultura de precisión	14
1.6 Necesidad de la investigación	14
1.7 Conclusiones Parciales.....	15
Capítulo 2: Metodologías, tecnologías y herramientas de desarrollo	16
2.1 Introducción	16
2.2 Aplicación Web.....	16

2.3	Herramientas y tecnologías.....	17
2.3.1	Plataforma: GeneSIG 2.0	17
2.3.2	Librería de interfaz de mapas: OpenLayers 2.12	18
2.3.3	Servidor de Base de Datos: PostgreSQL v9.3.....	18
2.3.4	Extensión para bases de datos espaciales. PostGIS.....	18
2.3.4	Lenguaje de Programación del lado del Servidor: PHP 5.4	19
2.3.5	Lenguaje de Programación del lado del Cliente: JavaScript.....	19
2.3.6	Framework de desarrollo del lado del cliente: ExtJs 3.0	19
2.3.7	AJAX	20
2.3.8	Entorno de Desarrollo Integrado (IDE): NetBeans 8.2.....	20
2.3.9	Servidor Web Apache v2.0	20
2.3.10	Servidor de Mapas: MapServer v5.4.1	21
2.3.11	Lenguaje de modelado y Herramienta CASE	21
2.3.12	Lenguaje Unificado de Modelado 2.0 (UML).....	21
2.3.13	Herramienta case: Visual Paradigm 8.0.....	22
3.4	Metodología de desarrollo de software.....	22
2.4.1	Agile UP y Agile UP-UCI.....	23
2.5	Conclusiones parciales.....	24
Capítulo 3: Análisis, diseño e implementación del sistema		25
3.1	Introducción	25
3.2	Especificación de los requisitos del sistema.....	25
3.2.1	Requisitos funcionales (RF).....	25
3.2.2	Requisitos no funcionales (RNF).....	28
3.3	Arquitectura del sistema.	30
3.3.1	Arquitectura del sistema: Basado en componentes.....	30
3.4	Modelo de datos.....	31

3.5	Modelo de diseño mediante diagramas de clases del diseño.....	32
3.5.1	Descripción de las clases del diagrama	33
3.5.2	Patrones de diseño	35
3.6	Modelo de despliegue	36
3.7	Diagrama de componentes	37
3.8	Pruebas de software.....	39
3.9.1	Pruebas de caja blanca	39
3.9.2	Pruebas de caja negra.....	40
3.9	Diseño de los casos de prueba	40
3.10	Resultados de las pruebas	44
3.11	Resumen de resultados	47
3.12	Conclusiones parciales.....	48
	Conclusiones.....	50
	Recomendaciones	52
	REFERENCIAS BIBLIOGRÁFICAS.....	53
	Anexos	57

Índice de figuras

Figura 1 Modelo de datos.....	31
Figura 2 Diagrama de clases del paquete Monitor de rendimiento del módulo de análisis.....	33
Figura 3 Diagrama de despliegue.	37
Figura 4 Diagrama de componentes.	38
Figura 5 Resultados de las pruebas.	48

Índice de tablas

Tabla 1: Fases de AUP-UCI.....	23
Tabla 2: Especificación de requisitos. RF 37: Monitoreo de rendimiento.....	25
Tabla 3: Especificación de requisitos. RF 38: Mostrar alerta.....	26
Tabla 4: Especificación de requisitos. RF 28: Insertar cultivo.....	27
Tabla 5: Clases de equivalencia.	41
Tabla 6: Diseño del caso de prueba al RF 37, método caja negra, resultado esperado: correcto. Clase de equivalencia 18C.....	41
Tabla 7: Prueba al RF 37, método caja negra, resultado esperado: incorrecto. Clase de equivalencia 18I-1.	42
Tabla 8: Prueba al RF 37, método caja negra, resultado esperado: correcto. Clase de equivalencia 18C.	44
Tabla 9: Prueba al RF 37, método caja negra, resultado esperado: incorrecto. Clase de equivalencia 18I-1.	45
Tabla 10: Pruebas de aceptación, pruebas alfa, RF 37.	46
Tabla 11: Pruebas de aceptación, pruebas alfa, RF 37. Resultados.	46

Introducción:

Las empresas son la célula principal de la economía mundial, la unidad básica de producción de bienes y servicios y a su vez la máxima responsable de la calidad y disponibilidad de estos. Entre los disímiles procesos comunes que se llevan a cabo en las empresas se encuentra el manejo y control de los recursos con que cuentan. Sin embargo, el desarrollo alcanzado y la creciente expansión del mundo, ha traído consigo la necesidad del manejo y control de mayores cantidades de datos. Por ello, cada vez se destinan más recursos para la creación de eficientes sistemas de información que permitan obtener el máximo provecho de dicha información.

En Cuba la agricultura representa uno de los eslabones fundamentales en la economía del país, es por eso que se destinan muchos recursos a su mantenimiento. En aras de aumentar la producción se trata de gestionar los medios con que se cuentan de la manera más eficaz posible. Actualmente, en este sector, existen problemas que afectan la calidad de los servicios que se deben brindar a la población; el incumplimiento de los planes de producción, de las fechas límites, de los estándares de calidad y la violación del presupuesto establecido, son solo algunos de estos problemas. Las entidades encargadas de manejar la base agrícola nacional deben mantener un control eficiente de los recursos con que cuentan para la producción de bienes, control que se dificulta a medida que crece el volumen de información con la que se debe tratar, esta información es de difícil análisis cuando se realiza de manera manual y esto trae consigo afectaciones en el proceso de toma de decisiones y pérdida del tiempo de reacción ante imprevistos. Además, la información se maneja de forma aislada, cada especialista tiene sus propios datos y en caso de necesitarse deben de solicitársele personalmente, no siempre existe heterogeneidad en la información, ya que la manipulación por varios individuos con diversas capacidades puede deteriorar la fiabilidad de esta.

Para un sector que es clave en el sustento de la nación tales afectaciones acarrearán consecuencias aún mayores en el resto de la infraestructura económica y graves afectaciones a la sociedad como consumidor final. Cuba, se encuentra inmersa en un proceso de transformaciones desde el punto de vista tecnológico, este proceso hace especial énfasis en la incorporación de las tecnologías de la información y las comunicaciones (TIC) a cada uno de los sectores socio-económicos del país. Una parte fundamental de este proceso de transformaciones lo juega el ingeniero en ciencias informáticas. *“El ingeniero en ciencias informáticas tiene como objeto de la profesión el proceso de informatización de la sociedad; entendiéndose como tal, la introducción, de forma gradual, masiva y planificada, de las tecnologías de la información y las comunicaciones en todas las esferas de la sociedad, con el objetivo de incrementar la eficiencia y eficacia*

en todos los procesos y en aras de lograr el aumento en la calidad de vida de los ciudadanos” (Ministerio de Educación Superior, 2014).

Un papel fundamental en este campo lo desempeña la Universidad de las Ciencias Informáticas (UCI), que es, a su vez, una entidad desarrolladora de *software*, este centro aporta avances tecnológicos en forma de productos, servicios y conocimientos. Entre sus áreas de acción se encuentra la implementación de SIG (*Sistemas de información geográfica*) para dar solución a problemas que requieran almacenar y consultar datos geográficos, destacándose la plataforma GeneSIG, diseñada como pilar funcional para el desarrollo de SIG soberanos. Esta plataforma ha sido personalizada en varias ocasiones con el fin de dar solución a diversas necesidades. Sin embargo, existen técnicas y tecnologías que son tendencia a nivel mundial para las cuales nunca ha sido personalizada la plataforma, entre estas técnicas y tecnologías¹ se encuentra la agricultura de precisión, y cuya aplicación podría ser un factor de peso en la economía nacional.

Producto de la necesidad social, a la que debe responder la profesión Ingeniero en Ciencias Informáticas, se plantea que la informatización de los procesos ha de aplicarse en todos los sectores y procesos de la nación. Debido a los problemas planteados anteriormente en el sector agropecuario cubano y a la no informatización de sus procesos se evalúa la posibilidad de incorporar las TIC a este sector como parte de su proceso de informatización.

Los Sistemas de Información Geográfica (SIG) *“se sitúan en este contexto como una categoría más dentro de los Sistemas de Información que se caracterizan por utilizar datos “localizados” o geo-referenciados según coordenadas relativas a un sistema común de referencia espacial”* (Gran Turismo, 2012). Un SIG no es más que una integración organizada de hardware, *software (aplicación informática)* y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. (Ramírez Gómez, 2011). Hoy día, los SIG han ampliado su utilidad práctica, incluyendo funcionalidades cada vez más avanzadas y novedosas, mejor manejo de los datos y siendo de gran ayuda para la toma de decisiones en diferentes esferas.

La Agricultura de Precisión corresponde a una estrategia de administración que utiliza las tecnologías de la información y las comunicaciones para recolectar datos útiles desde distintas fuentes con el fin de apoyar

¹ Conjunto de conocimientos propios de una técnica. | Conjunto de instrumentos, recursos técnicos o procedimientos empleados en un determinado campo o sector.

decisiones asociadas a la producción de cultivos (García, y otros, 2008). Requiere un conjunto de tecnologías formadas por el Sistema Global de Navegación por Satélite (GNSS), sensores e imágenes tanto satelital como aerotransportada, junto con Sistemas de Información Geográfica para estimar, evaluar y entender dichas variaciones. La información recolectada por estos sistemas puede ser empleada para evaluar con mayor precisión la densidad óptima de siembra, estimar la cantidad adecuada de fertilizantes o de otros insumos necesarios, y predecir con más exactitud el rendimiento y la producción de los cultivos. La agricultura de precisión tiene su basamento sobre la incorporación de tecnologías que permitan tener datos actualizados y precisos acerca de los factores que intervienen en el proceso agrícola. Es mediante estas tecnologías que se pueden tomar decisiones eficaces a la hora de enfrentarse a situaciones no previstas, tener un mayor control de la información necesaria para la toma de decisiones, así como un mecanismo de estimación.

Analizando lo antes planteado se determinó el siguiente **problema a resolver**: ¿Cómo mejorar los procesos de monitoreo y control que se realizan sobre los recursos y procesos en las entidades agrícolas cubanas? Para resolver dicho problema se propone como **objetivo de la investigación**: Desarrollar un Sistema de Información Geográfica para el Monitoreo y Control de Recursos Agrícolas en Cuba mediante técnicas de agricultura de precisión (SIGMCRA). Para cumplir con dicho objetivo se define como **objeto de estudio**: Los sistemas de información geográfica para la agricultura de precisión; siendo los sistemas de información geográfica para la agricultura de precisión en Cuba el área de incidencia o **campo de acción**.

Para guiar la investigación se definen **las preguntas de investigación siguientes**:

- ¿Qué elementos teórico-metodológicos permitirían sustentar una investigación científica respecto a la problemática planteada?
- ¿Cuáles son los problemas de monitoreo, control y rendimiento presentes en el sector agropecuario cubano?
- ¿Cuáles son las causas de estos problemas y cómo influyen en los procesos de las entidades agrícolas del sector?
- ¿Existen herramientas, tecnologías u otras soluciones que apoyen los procesos de monitoreo y control en el sector agropecuario del país?
- ¿Qué herramientas, tecnologías y técnicas se pueden aplicar en la elaboración de una solución a los principales problemas detectados?
- ¿Cómo solucionar los problemas relacionados con el monitoreo y el control presentes en el sector agropecuario cubano?
- ¿Cómo verificar que se ha dado solución a la problemática planteada?

Para dar cumplimiento al objetivo planteado anteriormente se definen a continuación **las tareas que rigen la investigación:**

- Definición de un marco metodológico que permita guiar la investigación y definir las bases para el proceso de obtención, síntesis y análisis de información en aras de generar resultados científicos.
- Identificación de los factores que influyen de manera negativa en el rendimiento agrícola en Cuba, a partir de un análisis de los procesos de monitoreo y control en el sector agropecuario cubano con el objetivo de determinar cuáles de estos pueden solucionarse.
- Identificación y análisis de los elementos que dan origen a los problemas que afectan los procesos de monitoreo y control en aras de comprender mejor el estado actual de la temática.
- Identificación y caracterización de herramientas, investigaciones o soluciones existentes que resuelvan parcial o completamente los problemas presentes en estos procesos para valorar su aplicación como solución.
- Evaluación de las tecnologías, técnicas y herramientas comúnmente empleadas en el desarrollo de soluciones de este tipo para su posible empleo en la propuesta actual.
- Elaboración de una propuesta de solución que permita monitorear y controlar eficazmente la información, los recursos y procesos agrícolas, solucionando así los problemas que se presentan en estos aspectos en las entidades agropecuarias del país.
- Evaluación de los resultados obtenidos en la investigación con el objetivo de verificar la calidad del proceso investigativo y la solución propuesta.

Métodos de investigación:

Toda investigación que se realice debe tener una estructura y un objetivo determinado, es común que los investigadores pierdan el camino a seguir una vez que se han sumergido en las grandes cantidades de información a tratar. A su vez es común también que los resultados no queden apropiadamente documentados debido al no establecimiento de pautas de investigación básicas acerca de ¿qué y cómo buscar la información? En la actualidad existen diversos métodos científicos probados que son de frecuente aplicación en la obtención de información valiosa y coherente en una investigación científica. Estos apoyan y orientan al equipo de investigadores. Estos se dividen en dos tipos, empíricos y teóricos (Frida, y otros, 2005), siendo estos, a grandes rasgos, los que se basan en el conocimiento y/o experiencia de expertos y aquellos que se basan en el análisis teórico-metodológico de información relevante a la investigación respectivamente. A continuación, se describen los métodos empleados:

Métodos teóricos

Analítico-Sintético: *"Está integrado por el desarrollo del análisis y la síntesis, mediante el cual se descompone un objeto, fenómeno o proceso en los principales elementos que lo integran para analizar, valorar y conocer sus particularidades, y simultáneamente a través de la síntesis, se integran vistos en su interrelación como un todo"* (Zayas Agüero, 2014). Este método plantea que la síntesis se produce sobre la base de los resultados previos al análisis. Se realiza un estudio y comprensión de todos los principios y definiciones relacionadas con los SIG y la agricultura de precisión, para describirlos y sintetizarlos teniendo en cuenta el objetivo de la investigación. Se utilizó para la definición de la base teórica mediante el análisis de la bibliografía y de las características principales de los procesos relacionados con la agricultura de precisión y los SIG con el objetivo de seleccionar los elementos principales para el desarrollo de este trabajo.

Análisis documental: *"El análisis documental constituye un proceso ideado por el individuo como medio para organizar y representar el conocimiento registrado en los documentos, cuyo índice de producción excede sus posibilidades de lectura y captura. La acción de este proceso se centra en el análisis y síntesis de los datos plasmados en dichos soportes mediante la aplicación de lineamientos o normativas de tipo lingüístico; a través de las cuales se extrae el contenido sustantivo que puede corresponder a un término concreto o a conjuntos de ellos tomados aisladamente, o reunidos en construcciones discursivas. Por consiguiente, su finalidad es facilitar la aproximación cognitiva del sujeto al contenido de las fuentes de información"* (Peña Vera, y otros, 2007). Este método es empleado mediante la recopilación de información asociada a los términos que son de interés y la selección de los elementos a profundizar con el objetivo de reducir el volumen de información sobre la cual realizar la síntesis y el análisis.

Métodos empíricos

Entrevista: *"La recogida de información a través de un proceso de comunicación, en el transcurso del cual el entrevistado responde a cuestiones, previamente diseñadas en función de las dimensiones que se pretenden estudiar, planteadas por el entrevistador"* (Puente, 2011). Se realizaron entrevistas semi-estructuradas al MsC. Yuniel Eliades Proenza Arias. Estas entrevistas se encuentran en un documento anexo llamado "Entrevista_Requisitos_Cliente.docx"

Para un correcto entendimiento de la investigación se decidió estructurarla en tres capítulos quedando distribuida la información de la siguiente manera:

Capítulo 1: Marco y base teórica: Abarca la base teórica de la investigación y los elementos claves a tener en cuenta durante la investigación.

Capítulo 2: Metodologías, tecnologías y herramientas de desarrollo: Se decide la metodología a emplear y se realiza un análisis de las herramientas y tecnologías a utilizar.

Capítulo 3: Análisis, diseño e implementación del sistema: Se diseña el modelo de implementación y se realizan las pruebas al sistema, se tienen además los resultados de estas.

Capítulo 1: Marco y base teórica

1.1 Introducción

En este capítulo se hace alusión a la base teórica de la investigación. Se especificarán los conceptos asociados al problema y los aspectos claves de la solución que se propone como respuesta al objetivo general. A partir de los elementos que serán expuestos en este capítulo se permitirá al lector alcanzar una mayor y mejor comprensión de cómo se pretende resolver la problemática planteada. Al finalizar este capítulo se hará un análisis de las soluciones existentes, explicando el tipo de solución que brindan con respecto al objeto de estudio planteado, esto se realizará con el fin de hacer notar la importancia de la investigación.

1.2 Conceptos asociados al Problema

Para facilitar el correcto entendimiento acerca de los temas que serán abordados durante la investigación serán descritos un conjunto de conceptos asociados al dominio del problema:

- **Agricultura de Precisión (AP)**

Se concibe agricultura de precisión como:

La agricultura de precisión es un conjunto de técnicas orientado a optimizar el uso de los insumos agrícolas (semillas, agroquímicos y correctivos) en función de la cuantificación de la variabilidad espacial y temporal de la producción agrícola (Bongiovanni, y otros, 2006).

“La agricultura de precisión es considerada como un Sistema Alternativo Sostenible utilizado en la producción agropecuaria, mediante el cual se emplean diferentes métodos o herramientas tecnológicas, como por ejemplo, los Sistemas de Posicionamiento Global (GPS) y la Electrónica, con el propósito de recopilar información en tiempo real sobre lo que sucede o puede suceder en los suelos y en los cultivos, para proceder de esa forma a la toma de decisiones en el futuro, que permitan el incremento de los rendimientos, la disminución de los costos de producción y la reducción de los impactos ambientales” (Programa corporativo para el desarrollo tecnológico agroalimentario y agroindustrial del cono sur, 2014).

Se puede definir entonces la agricultura de precisión como un conjunto de técnicas, herramientas y tecnologías enfocadas a incorporar sobre la base de la agricultura tradicional un mecanismo preciso para la obtención y representación de datos e información útil, mantener un control sobre los procesos agrícolas, determinar factores de riesgo, producción esperada/obtenida, ganancia/pérdida, estimación del rendimiento/aprovechamiento, apoyar los procesos de toma de decisiones, entre otras. Debido a su amplio

campo de acción no siempre es posible, necesario o recomendado emplearla a su máximo potencial, especialmente cuando no se necesita tanta información o cuando no se tienen demasiados recursos para la completa implementación. Por eso, aunque la agricultura de precisión puede ser un término abarcador, es posible encontrarse casos que apliquen solo una parte de esta y aún así encontrarse ante un sistema de agricultura de precisión. Está dividida en un conjunto de etapas, siendo estas:

Ciclo y etapas de la AP:

Los agentes involucrados en el desarrollo y adopción de las prácticas de la AP suelen dividir este conjunto de tecnologías en tres etapas diferentes (García, y otros, 2008 pág. 107):

- Recolección de datos (monitoreo de cultivo y suelos, mapas de producción, sensores).
- Procesamiento e interpretación de la información (análisis, sistemas informáticos).
- Aplicación de insumos (manejo variable, aplicación de correctivos).

Las primeras dos etapas se refieren a la obtención de información y a su análisis respectivamente, con el objetivo de aplicar correcciones y predicciones en el ámbito productivo de las organizaciones, están estrechamente relacionadas con los sensores, los cuales recaban información, y los sistemas de información geográfica que son los que poseen los componentes necesarios para realizar los análisis y mostrar los datos al usuario. La última etapa, se refiere principalmente a la estrecha relación que existe entre los procesos de análisis y la capacidad que tienen los dispositivos inteligentes (Smart devices en inglés) para coordinar sus acciones con el sistema y llevar a cabo las tareas necesarias.

Debido a la carencia de dispositivos inteligentes y la mayoría de los sensores requeridos por la AP, se deben de definir alternativas que permitan la obtención, de al menos, los datos más importantes, así como la aplicación de las técnicas de procesamiento e interpretación que no dependan estrictamente de aquellos datos que son actualmente in-adquiribles. Es por esto que se decide aplicar, de las etapas de la agricultura de precisión, solo las dos primeras, teniendo en cuenta además variaciones en la forma de obtener y analizar la información con el objetivo de obtener una verdadera aplicabilidad de la investigación propuesta.

• **Sistema de Información Geográfica (SIG)**

La información geo-referenciada, está presente, cada vez más, en las últimas tendencias de desarrollo. No sólo porque la disponibilidad de mapas digitales actualizados de cualquier lugar del planeta sea mayor, sino porque el hombre tiene a su disposición potentes herramientas informáticas que hacen un uso intensivo de esta información para múltiples tareas en todo tipo de entornos y dispositivos.

Al igual que la propia geografía, es difícil definir el término SIG ya que engloba áreas muy diversas. Por esto no existe una única definición totalmente consensuada. Podría definirse a un SIG como un sistema de hardware, *software* y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados, para resolver problemas complejos de planificación y gestión (Ramírez Gómez, 2011). A continuación, una descripción breve de los conceptos relacionados con los SIG.

- **Información Geográfica**

“Se denomina Información Geográfica a aquellos datos espaciales geo-referenciados requeridos como parte de las operaciones científicas, administrativas o legales. Dichos datos espaciales suelen llevar una información alfanumérica asociada.” (Gutiérrez Puebla, y otros, 1994). El concepto de información geográfica no ha cambiado mucho, sigue siendo, en esencia, aquella información que como parte de sus datos incluye componentes geoespaciales de ubicación o localización, o sea aquella que se refiere a datos geográficos específicos de algo.

- **Datos espaciales**

Los datos espaciales son información sobre la localización y las formas de un objeto geográfico, representan la ubicación y la forma de estos y pueden incluir las relaciones entre con otros objetos geográficos, la topología, coordenadas entre otros. Se refieren a entidades o fenómenos que cumplen los siguientes principios básicos (Rigaux, y otros, 2002):

- Tienen posición absoluta sobre un sistema de coordenadas (x, y, z).
- Tienen una posición relativa frente a otros elementos del paisaje (topología, incluido, adyacente, cruzado, etc.).
- Tienen una figura geométrica que las representan (punto, línea, polígono).
- Tienen atributos que lo describen (características del elemento o fenómeno).

- **Cartografía:**

La cartografía (del griego *chartis*, mapa y *graphein*, escrito) es la ciencia que se encarga del estudio y de la elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales. Por extensión, también se denomina cartografía a un conjunto de documentos territoriales referidos a un ámbito concreto de estudio. (RAE)

La cartografía en la época de la Web 2.0 se ha extendido hasta Internet. Este término implica que existan

mapas creados de la manera tradicional mediante contribuciones de varios cartógrafos individuales o con información aportada por el público. En la actualidad, son numerosos los portales que permiten visualizar y consultar mapas de casi todo el mundo.

1.3 Elementos básicos de un SIG

Un SIG particulariza un conjunto de procedimientos sobre una base de datos con información georeferenciada o de tipo espacial. Su mayor utilidad está íntimamente relacionada con la capacidad que posee de construir modelos o representaciones del mundo real, esto se logra aplicando una serie de procedimientos específicos que generan información para el análisis (Moreno, y otros, 2014).

Es un sistema capaz de mantener y usar datos con localizaciones exactas en una superficie terrestre, una herramienta de análisis de información la cual debe tener una referencia espacial. Estos sistemas compuestos por hardware, *software* y procedimientos diseñados, permiten resolver problemas complejos de planeación y administración, cuentan con un grupo de elementos los cuales son (Membrides Espinosa, y otros, 2012):

Puntos: los puntos son elementos representados por un vector de coordenadas (x_1, x_2, x_3, x_n) .

Líneas: las líneas están formadas por un punto inicial y otro final, y pueden tener o no vértices entre sus dos puntos finales. Las líneas que no tienen vértices entre sus puntos finales se denominan como líneas, líneas individuales, líneas de dos puntos o líneas discretas. Las líneas con uno o más vértices entre sus puntos finales se conocen como polilíneas.

Curvas: las curvas son elementos circulares almacenados como polilíneas en la base de datos, la cual mantiene información sobre su radio y otros atributos de la curva. Las curvas pueden ser círculos completos o solo arcos y pueden definirse en el sentido que giran las agujas del reloj o en el sentido inverso.

Polígonos: los polígonos representan figuras geométricas cerradas, formadas por tres o más vértices que para simbolizarse utilizan colores sólidos y tramados. Aun cuando las polilíneas y las curvas pueden formar figuras cerradas, no conforman polígonos.

Rasters: celdas (en una matriz o una retícula, que en pantalla corresponde a los píxeles).

1.4 Representar Datos en un SIG

Los SIG poseen dos formas de representar los datos espaciales, a través de datos vectoriales o de datos raster (Membrides Espinosa, y otros, 2012).

El modelo raster supone la existencia de un área de estudio sobre la cual se sobrepone un sistema de

cuadrículas, donde cada unidad se denomina celda y tienen la misma forma. La información se representa mediante celdas que describen que se encuentra en ese punto. En este modelo, la captura de la información se realiza mediante los siguientes medios: scanners, imágenes de satélite, fotografía aérea, cámaras de video entre otros.

El modelo vectorial tiene dos elementos, la descripción de la imagen por medio de una lista de coordenadas, y una lista o tabla de las variables que se encuentran unidas a esa imagen. La información se representa mediante formas geométricas (puntos, líneas o polígonos). En este modelo, la captura de la información se realiza mediante: sistemas de geo posicionamiento global (GPS), entrada de datos alfanumérica, entre otros.

La elección de un modelo u otro dependerá de si las propiedades topológicas son importantes para el análisis. Sí es así, el modelo de datos vectorial es la mejor opción, pero su estructura de datos, aunque muy precisa, es mucho más compleja y esto puede ralentizar el proceso. Por ello, si el análisis de interés no requiere acudir a las propiedades topológicas, es mucho más rápido, sencillo y eficaz el uso del formato raster.

También es más por una estructura de datos vectorial cuando fácil decantarse hay que reflejar más de un atributo en un mismo espacio. Usar un formato raster obligaría a crear una capa distinta para cada atributo.

1.4.1 Ventajas del Modelo Raster

- Utiliza una estructura de datos muy simple.
- Las superposiciones de las diferentes capas se implementan de forma rápida y eficiente.
- Permite una forma más eficiente de representación cuando la variación espacial es muy alta. El modelo raster es muy apropiado para el tratamiento de imágenes de satélite.
- Da la posibilidad de generar modelos de elevación del terreno.

1.4.2 Desventajas del Modelo Raster

- La estructura de datos es menos compacta.
- Algunas relaciones topológicas son difíciles de representar.
- La mayoría de estos SIG se ven limitados por la cantidad de filas y columnas que pueden manejar, por tanto, la resolución dependerá de estas.

- La salida en duro no resulta de muy buena calidad.

1.4.3 Ventajas del Modelo Vectorial

- Posee una estructura de datos muy compacta.
- Codifica eficientemente la topología.
- La salida en papel presenta muy buenos productos.

1.4.4 Desventajas del Modelo Vectorial

- La estructura de datos es más compleja.
- Las sobreposiciones son más complejas de realizar.
- Si la variación espacial es baja, resulta poco eficiente la aplicación.
- El procesamiento de imágenes digitales no puede ser realizado eficientemente en este tipo de formato.

Se ha optado por seleccionar el modelo vectorial preferentemente sobre el modelo raster, entre otros motivos, por la necesidad de representar la topografía de los elementos geográficos a tener en cuenta, también debido a las limitaciones en las cantidades de filas y columnas que deben manejarse en el caso de emplear el modelo raster lo cual limitaría la resolución, además los resultados de impresión en un modelo vectorial son de mayor calidad.

1.3.4 Soluciones Existentes

Para la obtención de información relevante con respecto al desarrollo de un SIG para la agricultura en Cuba se decidió analizar algunas de las soluciones existentes actualmente en el mercado. Con el objetivo de determinar las características fundamentales de este tipo de *software*, así como también los recursos que necesitan estos para desplegarse. Además, se esperaba encontrar algún *software* que fuese aplicable a la problemática en cuestión. Entre las soluciones existentes analizadas se encuentran:

1.5.1 GeoAgro GIS

GeoAgro GIS es el primer *software* SIG desarrollado especialmente para la agricultura. Es ideal para ordenar la información disponible sobre los lotes, analizarla y confeccionar mapas que puedan ser interpretados por la maquinaria que realiza los trabajos de campo (sembradoras, fertilizadoras, monitores de rendimiento) (Corporation, TEK). Este *software* argentino cuenta con una calidad elevada y un amplio

soporte. La compañía que lo comercializa TEK² mantiene una constante innovación en el mismo y se ha convertido en una de las soluciones más eficaces del mercado, incorpora un conjunto de extras con la compra que lo convierten en una excelente elección. Entre sus funcionalidades más notables se encuentra la integración que tiene con la maquinaria de campo, permitiendo incluso el establecimiento de forma remota de tareas, también realiza análisis antes, durante y después de cada uno de los procesos de agricultura fundamentales, como, por ejemplo, la siembra, irrigación la fumigación y la recolección. La compra del mismo incluye además abundante soporte al usuario, el cual puede adquirir información no solo de la herramienta sino también de técnicas de producción y gestión que le permitan aprovechar al máximo el terreno. Dicho esto, cabe notarse su naturaleza comercial y privativa, además de la dependencia de tecnologías altamente avanzadas e innovadoras con las cuales no se cuentan. Además, al ser un *software* extranjero se encuentra en contraste con las políticas informacionales actuales del país.

1.5.2 PixAgri

PixAgri ha sido desarrollado por Spot Image³. Se trata de una oferta completa, disponible a nivel mundial, que permite a los profesionales del mundo agrícola gestionar con la máxima precisión sus prácticas y los insumos con el objetivo de alcanzar una rentabilidad óptima, con mapas de observación de parcelas, que ofrecen información sobre el estado de los cultivos que le ayudan a tomar decisiones. PixAgri ofrece la posibilidad de detectar de forma precoz la variabilidad intra-parcelaria e inter-parcelaria, para guiar mejor las decisiones. Es una herramienta de ayuda a la decisión que facilita, tanto a los productores como a los actores económicos, seguir el estado de los cultivos a escala de parcela y de región. Esta herramienta permite tomar decisiones en las etapas clave del desarrollo basadas en el estado de los cultivos y de la variabilidad observada. Tiene en cuenta factores tales como el crecimiento y la propagación de entidades externas a los cultivos para predecir eventos que los afecten. Emplea los últimos avances en tecnologías de observación satelital y este es su punto de fortaleza. Aunque es relativamente joven en el mercado ha incorporado rápidamente funcionalidades que le permiten hacer frente a algunas de las soluciones más antiguas que existen. Actualmente es una de las dos herramientas más utilizada en la agricultura de precisión en Argentina, el país líder en el mundo en este ámbito A pesar de sus logros esta solución, está altamente vinculada a satélites condición que la convierten en un pobre candidato a aplicar como propuesta de solución.

² Compañía desarrolladora de soluciones digitales que incorporan tecnologías de la informática y las comunicaciones a sistemas de información geográfica para la agricultura de precisión.

³ Es una empresa diseñadora de software para el tratamiento de información geográfica mediante el empleo de los satélites SPOT que orbitan alrededor de la tierra.

1.5.3 Sistema de información geográfica para la gestión de cultivos en Cuba

El sistema de información geográfica para la gestión de cultivos en Cuba (SIGGCC) es una solución nacional diseñada con el objetivo de brindar una herramienta que permitiese a los usuarios tener un control de los terrenos, insumos y cultivos en las entidades agrícolas en los que se desplegará. Es un sistema sencillo y eficiente que emplea los recursos de los que se dispone e intenta optimizar los procesos en base a las necesidades y posibilidades reales del país. Es una solución a la medida para el sector agropecuario que se desarrolló además sobre la plataforma GeneSIG de la Universidad de ciencias informáticas (UCI). A pesar de ser una herramienta con grandes potencialidades no cumple con las crecientes necesidades del país, la gestión y el control de los recursos no es el único objetivo planteado por el país en aras de aumentar la disponibilidad y reducir los costos. El análisis de la documentación asociada a esta herramienta ha sido revelador para los involucrados en el proceso de desarrollo, define de manera concisa y clara algunas de las funcionalidades que se esperarían de este tipo de sistemas.

1.5.4 Sistema de información geográfica para la agricultura de precisión

SIGAP, sistema de información geográfica para la agricultura de precisión, es otra de las respuestas dadas por la UCI a las crecientes necesidades del país en el sector agrícola. Desarrollada también sobre la plataforma GeneSIG e incorporando algunos elementos básicos de la agricultura de precisión se plantea como una solución mejor al problema de la gestión y el control de los procesos agrícolas cubanos. Entre las funcionalidades más destacables del mismo se encuentra un sistema de alertas o notificaciones que guían algunos de los procesos que se realizan actualmente en los campos de cultivo. Estas notificaciones hacen notar su valor cuando se comprende su necesidad para reducir y/o eliminar el error o negligencia humana. Tanto así que se decide tomar algunos de los aspectos de este sistema e incorporarlos, de una forma mejorada, a la solución que se propone.

1.6 Necesidad de la investigación

Nuestro país promueve una política económica sobre la base del respeto al medio ambiente, la renovabilidad y el desarrollo sostenible. Estas políticas entran en contraste con algunas de las prácticas consumistas aplicadas en países capitalistas. Además, las condiciones particulares de la nación en términos de disponibilidad de recursos y tecnologías, así como también el clima y la topografía hacen las soluciones existentes, como mínimo, no óptimas. Si adicionamos a esto la búsqueda de soberanía tecnológica por la que aboga el país queda claro que no es factible la implantación de un sistema extranjero. Aunque las soluciones nacionales sí comprenden los elementos abordados anteriormente, no proveen una solución definitiva a los problemas que se presentan actualmente en el sector agropecuario cubano, podrían considerarse soluciones parciales en el mejor de los casos. Aunque su análisis permitió al equipo de

desarrollo arribar a conclusiones importantes se ha decidido que es necesario un nuevo sistema informático que responda a las necesidades que se plantean.

1.7 Conclusiones Parciales

Con el desarrollo de este capítulo se ha logrado definir de forma clara todos los aspectos teóricos y conceptuales, necesarios para la realización de la investigación propuesta y se ha realizado un análisis de algunas soluciones existentes con el fin de obtener un punto de partida sobre el cual basar la investigación. Se alcanza una comprensión de la eficacia de los sistemas de información geográfica como herramientas de apoyo a la toma de decisiones y de la agricultura de precisión como tecnología de apoyo a los procesos agrícolas en aras de aumentar el rendimiento y el control que se posee sobre los insumos y procesos, las tecnologías que emplea y los procesos que realiza. Se decide la necesidad de una nueva solución informática que responda a los problemas que existen en las entidades agrícolas nacionales y se arriba a la conclusión de que la mejor forma de dar respuesta a las necesidades existentes es mediante el desarrollo de un sistema de información geográfica que permita monitorear y controlar los procesos y recursos agrícolas mediante la incorporación de técnicas de agricultura de precisión. A partir del resultado del estudio aquí plasmado, se puede proceder a seleccionar las características que debe brindar este sistema y seleccionar las tecnologías necesarias para el desarrollo del componente de *software*.

Capítulo 2: Metodologías, tecnologías y herramientas de desarrollo

2.1 Introducción

En este capítulo se hace un análisis de las herramientas y tecnologías a ser utilizadas. Se abarcan las principales características de la Plataforma GeneSIG y se describen las tecnologías y herramientas que la componen, así como otras herramientas de desarrollo y diseño seleccionadas. Además, se realiza una descripción de la metodología seleccionada para el desarrollo del *software*. Esto se realiza con el objetivo de definir los procesos a llevar a cabo, en cuantas etapas ha de ser, artefactos a generar entre otros elementos; además el análisis de las tecnologías y herramientas de desarrollo permite determinar cómo han de ser las características del ambiente de desarrollo, conocimientos que debe dominar el equipo de trabajo y otros aspectos relacionados a las potencialidades y limitaciones de dichas herramientas y tecnologías.

2.2 Aplicación Web

Los sistemas Web se han convertido en una tendencia hoy en día en el mercado informático en cuanto a soluciones hechas a la medida, tanto generales como específicas para pequeñas, grandes y medianas organizaciones. Esta tendencia se debe principalmente a sus ventajas, las cuales se describen a continuación:

Multiplataforma: al no requerir más que un navegador web, estas aplicaciones son transparentes en cuanto a la plataforma desde la cual se haga uso del sistema, no existen restricciones en cuanto a sistema operativo, licencias, hardware o *software*.

Fácil de actualizar: teniendo en cuenta que el sistema se encuentra centralizado y ubicado en un servidor web al cual acceden los usuarios, permite que sea fácil su actualización, de modo que una vez que se actualice la aplicación central se garantiza que todos los usuarios que accedan a ella hagan uso de la última versión del sistema y que los datos estén correctamente actualizados.

Debido a estas ventajas y a las características *in situ* (en el lugar) de los procesos agrícolas se hace necesario tener toda la información necesaria centralizada y disponible para su consulta por todos los involucrados en el proceso desde cualquier lugar desde el que se tenga acceso. Además, el intercambio de datos geo-espaciales a través de tecnologías web resulta muy conveniente no solo por la eficiencia del protocolo HTTP (*Protocolo de transferencia de hipertexto o Hypertext transfer protocol en inglés*) sino también por las ventajas visuales y de interfaz que presentan este tipo de soluciones informáticas.

2.3 Herramientas y tecnologías

2.3.1 Plataforma: GeneSIG 2.0

GeneSIG surge por la necesidad que se tiene, de contar con un producto soberano que sirva como soporte al desarrollo de aplicaciones de SIG en entornos libres, en el país. Además de su objetivo fundamental que es realizar la representación geoespacial de la información, se creó aprovechando las ventajas que ofrece el *framework* (*plataforma o marco de trabajo*) CartoWeb, permitiendo la creación de un SIG adaptable a cualquier situación.

Dicha plataforma posee un grupo de ventajas funcionales (Membrides Espinosa, y otros, 2012), siendo las siguientes:

- Representación geoespacial de la información asociada a negocios específicos.
- Servicios de acceso a la información geográfica, para su consulta, análisis y visualización, mediante una interfaz sencilla y de fácil manejo que pueda ser utilizada por usuarios no especializados.
- Actualización de las bases de datos a través de un mecanismo de réplica consistente.
- Integración con la información raster existente (imágenes de satélite, orto-fotos o mapas escaneados) con información vectorial.

La plataforma GeneSIG es una herramienta que permite realizar representaciones y análisis de información referenciada geográficamente (Varen Caballero, 2010). Posee una estructura basada en componentes, lo que la convierte en una plataforma altamente escalable. De igual modo permite separar los servidores de bases de datos y web en dos estaciones de trabajo diferentes, balanceando la carga del sistema, aumentando su disponibilidad y disminuyendo la posibilidad de fallas.

GeneSIG incluye un módulo para el tratamiento de la información teniendo en cuenta criterios de seguridad definidos en roles y permisos sobre los recursos que maneja. Producto de su arquitectura basada en componentes la personalización del mismo puede aplicarse a cualquier negocio que lo requiera a través de la reutilización. Brinda servicios de georreferenciación y localización, así como la inclusión de datos y ubicación de nuevos objetos (Membrides Espinosa, y otros, 2012).

Cumple y respeta la estructuración y el diseño propuestos por CartoWeb. Presenta un abanico bastante completo de características propias de un geoportal, con posibilidad de ir añadiendo o desarrollando nuevos *plugins* (complementos). Y es precisamente a través de estos *plugins*, que GeneSIG posee un amplio conjunto de funcionalidades, que actúan como herramientas de esta.

La plataforma GeneSIG posee una arquitectura distribuida, empleando como base cartográfica una información certificada por especialistas que laboran en su desarrollo, y sobre ella un conjunto de objetos representados geo-espacialmente que contienen información asociada. El sistema tiene una arquitectura cliente-servidor, orientada a objetos y basada en componentes. Su utilización proveería al equipo de desarrollo de una plataforma sólida y bien definida que incorpora muchas funcionalidades propias de los SIG, ahorrando tiempo y esfuerzo y aumentando la calidad del producto final.

2.3.2 Librería de interfaz de mapas: OpenLayers 2.12

OpenLayers es una biblioteca de JavaScript de código abierto para mostrar mapas interactivos en los navegadores web. Ofrece un API (*Interfaz de programación de aplicaciones o application programming interface en inglés*) para acceder a diferentes fuentes de información cartográfica en la red: Servicios de mapa en la web (*Web Map Services en inglés*), Mapas comerciales, distintos formatos vectoriales (Open Source Geospatial Foundation).

Es un visor de mapas en JavaScript, una librería del lado del cliente, la descarga de estos se realiza directamente desde el navegador a través de AJAX; no genera tráfico en el servidor de aplicaciones, los mapas se descargan directamente del servidor de mapas que suele ser una pieza diferenciada a nivel de sistemas.

2.3.3 Servidor de Base de Datos: PostgreSQL v9.3

Es un sistema de administración de base de datos objeto-relacional de código abierto que ofrece muchas características modernas, liberado bajo la licencia BSD. Puede ser extendido por el usuario en múltiples formas, usado, modificado y distribuido libre de cargos para cualquier propósito, sea privado, comercial o académico.

“Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos” (Group, Global Development, 2008).

2.3.4 Extensión para bases de datos espaciales. PostGIS

PostGIS es un módulo que añade soporte para objetos geográficos a las bases de datos objeto-relacional PostgreSQL, convirtiéndola en bases de datos espaciales. Incorpora un conjunto de características como son: datos espaciales, índices espaciales y funciones para operar con ellos (Open Source Geospatial

Foundation). Es un módulo Open Source (*código abierto*) que cumple con los estándares definidos por la OGC (*Consortio Geoespacial de código Abierto u Open Geospatial Consortium* en inglés). PostGIS es un producto muy popular hoy día y posee una muy buena integración con herramientas FOSS (*Software libre y de código abierto o Free and Open Source Software* en inglés).

2.3.4 Lenguaje de Programación del lado del Servidor: PHP 5.4

PHP es un lenguaje de programación interpretado, usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa " Hypertext Preprocessor" (inicialmente PHP Tools, o, Personal Home Page Tools). Actualmente también se puede utilizar para la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. Este lenguaje unido a un script CGI (Common Gateway Interface) permite procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y soporta la mayoría de los servidores web.

2.3.5 Lenguaje de Programación del lado del Cliente: JavaScript

JavaScript es un lenguaje de script multiplataforma orientado a objetos. Diseñado más bien para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos. El núcleo de JavaScript contiene un conjunto central de objetos, tales como Array (arreglos), Date (fechas) y Math (objetos matemáticos). Además de un conjunto central de elementos del lenguaje tales como los operadores, estructuras de control y sentencias.

Nació con la necesidad de permitir a los desarrolladores de sitios webs crear páginas que permitan intercambiar con los usuarios, porque se necesitaba crear sitios webs de mayor complejidad. Al ser un lenguaje del lado del cliente este es interpretado por el navegador, no se necesita tener instalado ningún *framework* (Mozilla).

2.3.6 Framework de desarrollo del lado del cliente: ExtJs 3.0

ExtJs es una librería JavaScript que permite construir aplicaciones web complejas además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. Esta librería incluye: Componentes UI de alto rendimiento y personalizables, un modelo de componentes extensibles, un API fácil de usar, licencias Open Source (GPL) y comerciales. Este *framework* cuenta con un conjunto de componentes (widgets) para incluir dentro de una aplicación web, como son cuadros y áreas de texto, campos para fechas, campos numéricos,

combos, radio-buttons y check-boxs, entre muchos otros. Varios de estos componentes están dotados de comunicación con el servidor usando AJAX. También contiene numerosas funcionalidades que permiten añadir interactividad a las páginas HTML (Sencha).

2.3.7 AJAX

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Application" publicado por Jesse James Garrett el 18 de febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo. En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML". Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas (W3C).

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

2.3.8 Entorno de Desarrollo Integrado (IDE): NetBeans 8.2

"NetBeans es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso" (Community, NetBeans)

Es una base modular y extensible usada como una estructura de integración para crear grandes aplicaciones de escritorio, web y de muchos otros tipos. Le permite al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están la administración de las interfaces de usuario, de las configuraciones, administración del almacenamiento y de ventanas, además de *framework* basados en asistentes, entre otros.

2.3.9 Servidor Web Apache v2.0

Corre en varios sistemas operativos, lo que lo hace prácticamente universal. Apache es una tecnología gratuita de código fuente abierto. Altamente configurable, de diseño modular y es muy sencillo ampliar sus capacidades. Actualmente existen muchos módulos que son adaptables a este, y que pueden ser instalados cuando se necesiten. Otro elemento importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada. También trabaja con Java y páginas java script. Teniendo todo el soporte que se necesita para tener páginas

dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. Facilita la creación de ficheros de log, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. Se encarga de gestionar la mayor parte de las funciones de lógica de negocio y de acceso a los datos de la aplicación (Apache Software Foundation).

2.3.10 Servidor de Mapas: MapServer v5.4.1

Es una aplicación CGI (Common Gateway Interface) de carácter libre desarrollada para construir aplicaciones que sirvan mapas a través de Internet, la cual corre bajo plataformas Linux/Apache, Windows XP/NT/98/95. Es calificado como el producto de *software* libre más maduro y popular para el desarrollo de aplicaciones web espaciales. Programado en el lenguaje C, accede a todos los formatos soportados por GDAL (Geospatial Data Abstraction Library), y permite programación con PHP y otros muchos lenguajes. La información visualizada por el servidor son archivos “.shape”, la cual resulta en la generación de lado del servidor web de imágenes estáticas (JPEG, GIF, PNG, etc.) debido a las peticiones realizadas por los clientes (Open Source Geospatial Foundation).

2.3.11 Lenguaje de modelado y Herramienta CASE

Las herramientas de modelado forman un componente importante en el entorno de desarrollo de *software*, pues son esenciales para el análisis de sistemas, mejoran la forma en que ocurre el desarrollo e influyen sobre la calidad del resultado final. En la actualidad existen una serie de herramientas con este fin, entre ellas destacan: Visual Paradigm y Rational Rose Enterprise Edition, las que usan el lenguaje de modelado UML para la especificación, la visualización, la construcción y la documentación de los artefactos de los sistemas de *software*. La documentación de un sistema es tan importante como el sistema en sí, no solo lo describen para desarrollar futuras versiones, sino que certifican que el proceso de desarrollo fue correctamente implementado y controlado certificando también un mayor grado de profesionalidad y calidad al mismo. De ahí que la selección del lenguaje de modelado y la herramienta CASE conlleve una alta significación para el proyecto.

2.3.12 Lenguaje Unificado de Modelado 2.0 (UML)

“UML (por sus siglas, del inglés Unified Modeling Language) es el lenguaje estándar de modelado para software. Es un lenguaje gráfico que visualiza, especifica, construye y documenta los artefactos del sistema. UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados, proporcionando a los desarrolladores un vocabulario que incluya tres categorías: elementos, relaciones y diagramas. UML suministra mecanismos de extensibilidad, los cuales permiten a

sus usuarios clarificar su sintaxis y su semántica. Puede ajustarse a un sistema, proyecto o proceso de desarrollo específico si es necesario.” (Jacobson, y otros, 2000)

UML soporta de forma innata casi todas las metodologías de *software* actuales y desde el año 2005, es un estándar aprobado por la ISO como ISO/IEC 19501 y en el año 2012 se actualizó la norma a la última definitiva. Soporta gran variedad de diagramas entre ellos estructurales, de comportamiento y de interacción, es uno de los lenguajes más utilizados actualmente, y esto se debe en gran parte a su extensibilidad, la cual le permite cubrir cualquier necesidad de modelado mediante la incorporación de elementos redefinidos o totalmente nuevos. El hecho de ser un lenguaje estandarizado y altamente eficiente lo convierten en candidato más que suficiente a selección; además el equipo de desarrollo se encuentra capacitado en el mismo por lo que se ha determinado como lenguaje de modelado del sistema.

2.3.13 Herramienta case: Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE que utiliza “UML”: como lenguaje de modelado. Además, soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permitiendo una rápida construcción de aplicaciones de calidad. Utilizando esta herramienta se pueden dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Y aunque se caracteriza por la utilización de UML como lenguaje de modelado es altamente escalable y soporta la incorporación de muchos otros lenguajes (VisualParadigm).

Se caracteriza por:

- Ingeniería inversa - Código a modelo, código a diagrama.
- Generación de código - Modelo a código, diagrama a código.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Permite crear diagramas de flujo de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.

3.4 Metodología de desarrollo de software

“Una metodología de desarrollo de sistemas se refiere al armazón que se usa para estructurar, planificar, y controlar el proceso de desarrollar un sistema de información. Una amplia variedad de tales armazones ha evolucionado a través de los años, cada uno con sus debilidades y fortalezas. Una metodología de desarrollo no es, necesariamente, aplicable a todo tipo de proyectos. Cada una de las metodologías es la

más indicada para ciertos tipos específicos de proyectos, basándose en consideraciones técnicas, organizacionales, del proyecto y del equipo” (Services, Centers for medicare & medicaid, 2005).

“Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto, pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y final”. (Medina, 2015)

En el mundo existen muchas definiciones sobre lo que es una metodología. De una forma u otra todas ellas coinciden en que debería tener al menos las siguientes características:

- “Define como se divide un proyecto en fases y las tareas a realizar en cada una”.
- “Para cada una de las fases está especificado cuales son las entradas que reciben y las salidas que producen”.
- “Tienen alguna forma de gestionar el proyecto”.

2.4.1 Agile UP y Agile UP-UCI

La Metodología AUP es una versión simplificada de RUP que aplica técnicas ágiles. Esta describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software*, realizando iteraciones dentro del proyecto y descomponiendo este en mini proyectos con el objetivo de tener un mayor control sobre cada una de las iteraciones (Sánchez Rodríguez, 2014).

En la UCI se ha diseñado una variación de AUP nombrada AUPUCI la cual es precisamente la que se define para los aplicativos que se diseñan sobre la plataforma GeneSIG. En AUPUCI las fases de elaboración, construcción y transición se unifican en una misma fase llamada fase de ejecución y se añade una fase llamada fase de cierre donde se analizan los resultados del proyecto y su ejecución. De esa forma quedan estipuladas 3 fases: inicio, ejecución y cierre.

Tabla 1: Fases de AUP-UCI

AUP-UCI	Descripción
Inicio	En esta fase se llevan a cabo las actividades relacionadas con la planificación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente, del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y por tanto un análisis de su factibilidad.

Ejecución	En esta fase se ejecutan las actividades: determinación de los requisitos funcionales y no funcionales, modelado del negocio y del sistema, implementación y transición.
Cierre	En esta fase se analizan los resultados del proyecto y se realizan las actividades formales de cierre del proyecto.

En cuanto al escenario a emplear se ha seleccionado el cuarto escenario; el cual modela sus requisitos según historias de usuario, este se aplica a los proyectos que hayan evaluado los procesos a informatizar y como resultado obtengan sus procesos muy bien definidos, además es el escenario que se encuentra definido para los proyectos que trabajen sobre la plataforma GENESIG.

2.5 Conclusiones parciales

En el capítulo se realizó una caracterización de la plataforma GeneSIG y de las herramientas y tecnologías que utiliza. Del mismo modo fueron expuestas un conjunto de características fundamentales de la metodología que va a ser utilizada (AUPUCI). Esto ha servido para seleccionar los elementos principales a tener en cuenta a la hora de pasar al diseño e implementación del sistema, artefactos a generar, pruebas a realizar y fases de desarrollo. Permitted comprender además los elementos a las que debe abordar en cada una de las etapas; el análisis a la plataforma ha permitido a los involucrados en el desarrollo del sistema comprender el enfoque del mismo, sus potencialidades y limitaciones, en aras de sacarle el máximo provecho. Además, la selección de GeneSIG permite al equipo de desarrollo, en cierto grado, abstraerse de algunos elementos relacionados con los SIG ya que esta implementa muchas funcionalidades propias de estos sistemas. Mediante su caracterización se comprenden de mejor manera algunos de los elementos relacionados al trabajo con información espacial y la representación de estos. Y comprende cómo ha de quedar diseñado, distribuido e implementado el mismo de una manera básica, lo cual facilita los procesos de modelado, diseño e implementación del sistema a los que se abordara en el próximo capítulo.

Capítulo 3: Análisis, diseño e implementación del sistema

3.1 Introducción

Como hemos visto anteriormente las metodologías generan un conjunto de salidas llamadas artefactos, estas salidas son tan importantes como la solución pues describen el proceso y permiten comprender el sistema. Estos artefactos permiten entender, modificar y respaldar todo el proceso de forma que futuras modificaciones sean más fáciles de realizar y con mayor calidad e integración con lo ya desarrollado, además, son constancia de la calidad del proceso de desarrollo. En el siguiente capítulo se pretende describir el sistema SIGMGRA a partir de la documentación generada en el proceso de desarrollo de *software* durante el ciclo de vida descrito en la metodología AUPUCI. Se abordan también los elementos fundamentales del análisis, diseño e implementación del sistema; los requisitos funcionales y no funcionales, principales artefactos y las pruebas realizadas.

3.2 Especificación de los requisitos del sistema

En los subsiguientes epígrafes se listan los requisitos funcionales y no funcionales que fueron identificados.

3.2.1 *Requisitos funcionales (RF)*

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Definen necesidades del cliente y muestran cómo serán llevados a la práctica las funcionalidades del sistema. En la siguiente sección solo se detallarán algunos de los requisitos funcionales más importantes, se puede acceder a la totalidad de los requisitos del sistema en el documento adjunto “Compendio de requisitos.docx” o en el anexo a este documento donde se encuentra una breve lista.

En la variante que se aplica de la metodología se emplean historias de usuario para modelar y formalizar los requisitos. Estas definen un requisito de manera breve y simple y son generalmente escritas por el propio cliente. A continuación, la especificación de los requisitos: Monitoreo de rendimiento, Mostrar alerta e Insertar cultivo.

Tabla 2: Especificación de requisitos. RF 37: Monitoreo de rendimiento.

Requisito: RF 37		Nombre del requisito: Monitoreo de rendimiento	
Programador: José Rubén Machado Vega		Iteración Asignada: 1	

Prioridad: Alta	Tiempo Estimado: 7 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Proceso mal implementado o con fallas de seguridad. • Débil validación de los datos relacionados al proceso. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección de esferas superiores • Planificación irreal. 	Tiempo Real:
Descripción: El sistema, dado un conjunto de datos, debe ser capaz de realizar un resumen informativo acerca de la producción, calidad, aprovechamiento y ganancia-perdida en un ciclo productivo. Para ello se necesitan los datos asociados al cultivo en cuestión y la entrada manual de algunos datos nuevos.	
Observaciones: <ul style="list-style-type: none"> • El sistema muestra como resultado un resumen informativo con varios elementos tales como: producción, calidad, aprovechamiento y ganancia-pérdida. 	
Prototipo de interfaz:	

Tabla 3: Especificación de requisitos. RF 38: Mostrar alerta.

Requisito: RF 38	Nombre del requisito: Mostrar Alerta
Programador: José Rubén Machado Vega	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Errónea sincronización del tiempo de las alertas o solapamiento múltiple de eventos importantes. • Errores en la verificación de las fechas. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección de esferas superiores • Planificación irreal. 	Tiempo Real:

Descripción:

El sistema debe permitir alertar a los usuarios sobre la proximidad de una fecha determinada ya sea para el periodo de recolección de un cultivo, siembra o la fertilización. Esta funcionalidad utilizará los datos de cada cultivo para alertar de la proximidad de eventos tales como la riego, fertilización y cosecha de una determinada parcela, así como también una recomendación del fertilizante a utilizar.

Observaciones:

- El sistema muestra como resultado una alerta para notificar de la cercanía de una fecha determinada.
- El sistema brinda una recomendación acerca del fertilizante a utilizar según criterios de peligrosidad del mismo al medio ambiente.

Prototipo de interfaz:

Tabla 4: Especificación de requisitos. RF 28: Insertar cultivo.

Requisito: RF 28	Nombre del requisito: Insertar cultivo
Programador: José Rubén Machado Vega	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Débil validación de los datos relacionados al proceso. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección de esferas superiores • Planificación irreal. 	Tiempo Real:
Descripción: <p>Con este requisito se quiere que el usuario pueda insertar un tipo de cultivo a la entidad. Para ello se necesitan los datos del cultivo los cuales deben ser proporcionados en el momento de la inserción.</p>	
Observaciones: <ul style="list-style-type: none"> • El sistema inserta dicho cultivo a la base de datos. 	
Prototipo de interfaz:	

3.2.2 Requisitos no funcionales (RNF)

Los requisitos no funcionales son cualidades o propiedades que el producto debe poseer. Pueden ser restricciones o características especiales que debe cumplir el producto.

Confiabilidad:

RNF 1. La aplicación podrá ser utilizada sólo por los usuarios registrados.

RNF 2. Las excepciones habrán de manejarse y mostrarse de manera adecuada garantizando que la información mostrada incluya los mensajes y códigos del error.

Licencia:

RNF 3. La licencia de las herramientas y tecnologías a emplear ha de ser libre según lo estipulado en la licencia GPL, LGPL o cualquier otra aceptada por la FSF (*Fundación del software libre o Free software foundation en inglés*).

Diseño e Implementación:

RNF 4. Después de 2 horas de uso del sistema el usuario no debe perder el flujo de navegación por la interfaz más de 1 vez por cada 4 horas de trabajo con el mismo.

RNF 5. Se requiere el uso de Extjs.

RNF 6. Se requiere PostgreSQL 9.3 como gestor de bases de datos.

RNF 7. Se requiere el uso de Apache como servidor web.

Soporte:

RNF 8. El sistema debe brindar los artefactos definidos por la metodología y el expediente de proyecto en aras de garantizar su futura actualización y modificación.

Hardware:

RNF 9. Para garantizar un buen funcionamiento del sistema, o sea cumplir con lo definido en el requisito no funcional de eficiencia y evitar fallos por insuficiente poder de cómputo, el servidor donde estará desplegado deberá cumplir con las siguientes características:

- Procesador con una velocidad de 3.00 GHz
- 2 GB de RAM

- Tarjeta de red con una correcta configuración de dicha interfaz para el acceso desde las PC clientes.
- Capacidad de almacenamiento a consideración, no menos de 100 GB.

RNF 10. Para garantizar un buen funcionamiento, o sea la correcta ejecución y visualización del sistema, por la parte del cliente debe poseerse:

- Una interfaz de red con conexión al servidor.
- Al menos 512MB de memoria RAM.
- Y al menos 1GHz de microprocesador.

Software:

RNF 11. Para el caso de las PC clientes se debe utilizar cualquier navegador que soporte las tecnologías antes mencionadas y los estándares definidos por el W3C.

RNF 12. Para el caso de la PC servidor, por motivos de diseño e implementación del sistema, debe poseerse:

- Sistema operativo GNU/Linux (Ubuntu o derivados directos) 14.04
- Servidor web apache v2.4.7 con módulo PHP v5.4
- Gestor de base de datos PostgreSQL v9.3 con la extensión PostGIS v2.1.
- Servidor de mapas MapServer v5.4.7 con extensión PHP MapScript.

Interfaz:

RNF 13. La interfaz de la aplicación debe seguir el diseño definido en la plataforma GENESIG, respetando sus colores, barras de herramientas y ubicación de las partes de la interfaz de usuario.

Usabilidad

RNF 14. Después de 6 horas de uso por parte del usuario del sistema el mismo no debe cometer más de 1 error por hora de explotación del mismo.

RNF 15. El sistema debe ser de entorno web para garantizar el acceso de manera uniforme e independiente de la plataforma.

Eficiencia

RNF 16. Tiempo de respuesta

El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento, este tiempo no debe exceder los 30 segundos. Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación, pero no debe exceder los 30 segundos.

3.3 Arquitectura del sistema.

La arquitectura de *software* no es más que la estructura de un sistema compuesta por un conjunto de elementos y las relaciones entre ellos. Define también la forma de organización y distribución de sus partes. La arquitectura de *software* es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un *software*, permitiendo a los programadores, analistas y todos los interesados compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del *software*. Estas dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Una arquitectura expresa un esquema de organización estructural esencial para un sistema de *software*, que consta de subsistemas, sus responsabilidades e interrelaciones (Universidad de las Ciencias Informáticas).

3.3.1 Arquitectura del sistema: Basado en componentes

Una arquitectura basada en componentes describe una aproximación de ingeniería de *software* al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos bien definidos.

El desarrollo de *software* basado en componentes constituye una aproximación del desarrollo de *software* que describe, construye y emplea técnicas de *software* para elaborar sistemas abiertos y distribuidos, mediante el ensamblaje de partes de *software* reutilizables. (Iribarne Martínez, 2003)

El estilo de arquitectura basado en componentes tiene las siguientes características:

- Es un estilo de diseño para aplicaciones compuestas de componentes individuales.
- Hace énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen interfaces bien definidas.
- Define una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades.

Un componente de *software* individual es un paquete de *software*, un servicio web, o un módulo que encapsula un conjunto de funciones relacionadas.

Todos los procesos del sistema son colocados en componentes separados de tal manera que todos los datos y funciones dentro de cada componente están semánticamente relacionados (justo como son la contención de clases). Debido a este principio, con frecuencia se dice que los componentes son modulares y cohesivos.

De una manera simple se puede decir que la idea principal de estos componentes es que la cohesión sea alta y el acoplamiento tan bajo como sea posible. De esta forma se modulariza el producto final y se aplica la filosofía de divide y vencerás. Los principios fundamentales cuando se diseña un componente es que este debe ser: reusable, sin contexto específico, extensible, encapsulado e independiente.

3.4 Modelo de datos

El modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí (Roca Díaz., y otros, 2012). Por tal motivo, se le considera un aspecto fundamental para el desarrollo de cualquier aplicación que necesite almacenar datos.

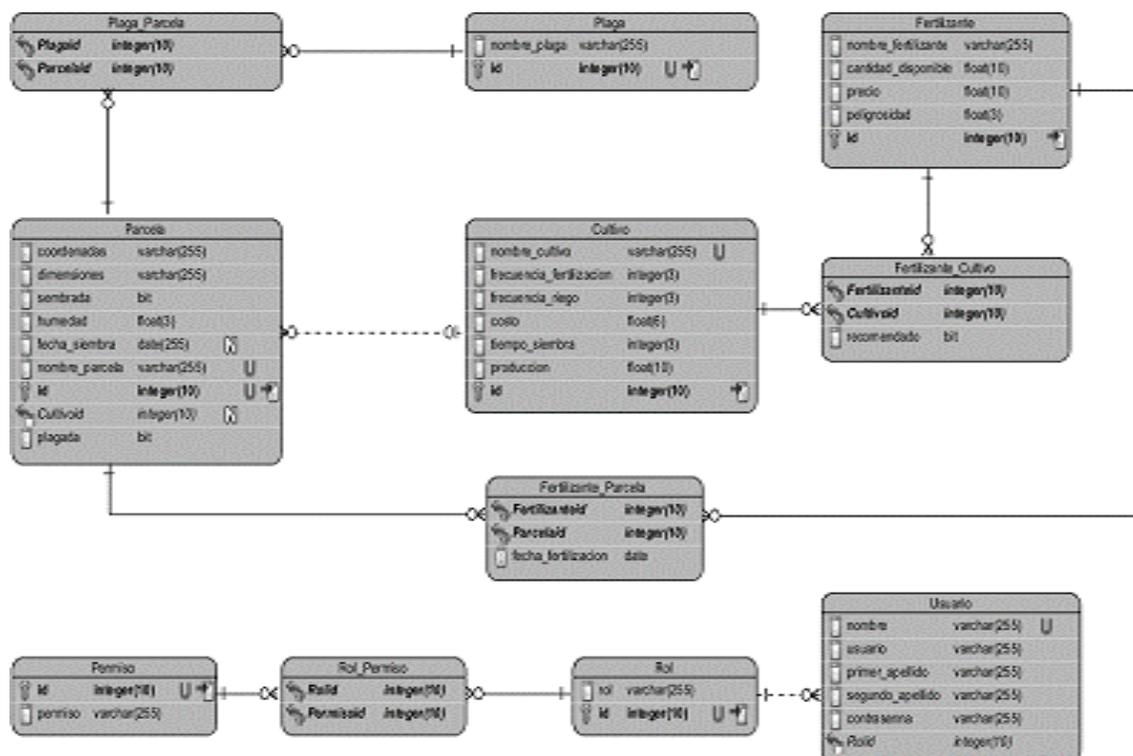


Figura 1 Modelo de datos.

Se pueden observar las entidades parcela, cultivo, plaga y fertilizante, las parcelas se refieren a las áreas de cultivo y poseen entre otros datos la ubicación geográfica (coordenadas) y las dimensiones; los cultivos hacen alusión a los productos que produce la entidad y comprende elementos tales como costo producción y frecuencia de fertilización. La relación entre parcelas y cultivos define que en un momento dado puede haber o no un cultivo sembrado en la misma, mientras que las parcelas solo pueden albergar un cultivo, un mismo cultivo puede encontrarse sembrado en más de una parcela. La entidad plaga hace referencia a las diferentes plagas que pueden afectar los terrenos. La relaciona de esta con las parcelas indican la presencia de una plaga en esa zona de cultivo. La entidad fertilizante posee datos de los mismos, tales como su costo y disponibilidad, la relación de este con los cultivos se utiliza con el fin de determinar que fertilizantes son aplicables a cada cultivo.

3.5 Modelo de diseño mediante diagramas de clases del diseño

El modelo de diseño es un modelo de objetos que describe la realización de las historias de usuario centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el producto final, es uno de los artefactos más importantes para describir la implementación del sistema y es intensamente usado por los programadores con dicho fin (Kruchten, 2004).

Entre los artefactos que genera el modelo de diseño se encuentra el Diagrama de Clases del Diseño donde se muestran las clases, interfaces y las relaciones entre ellas. A continuación, se expone un ejemplo de dichos diagramas, para acceder a la totalidad de diagrama de clases del diseño dirigirse al documento adjunto "Modelo de diseño.docx".

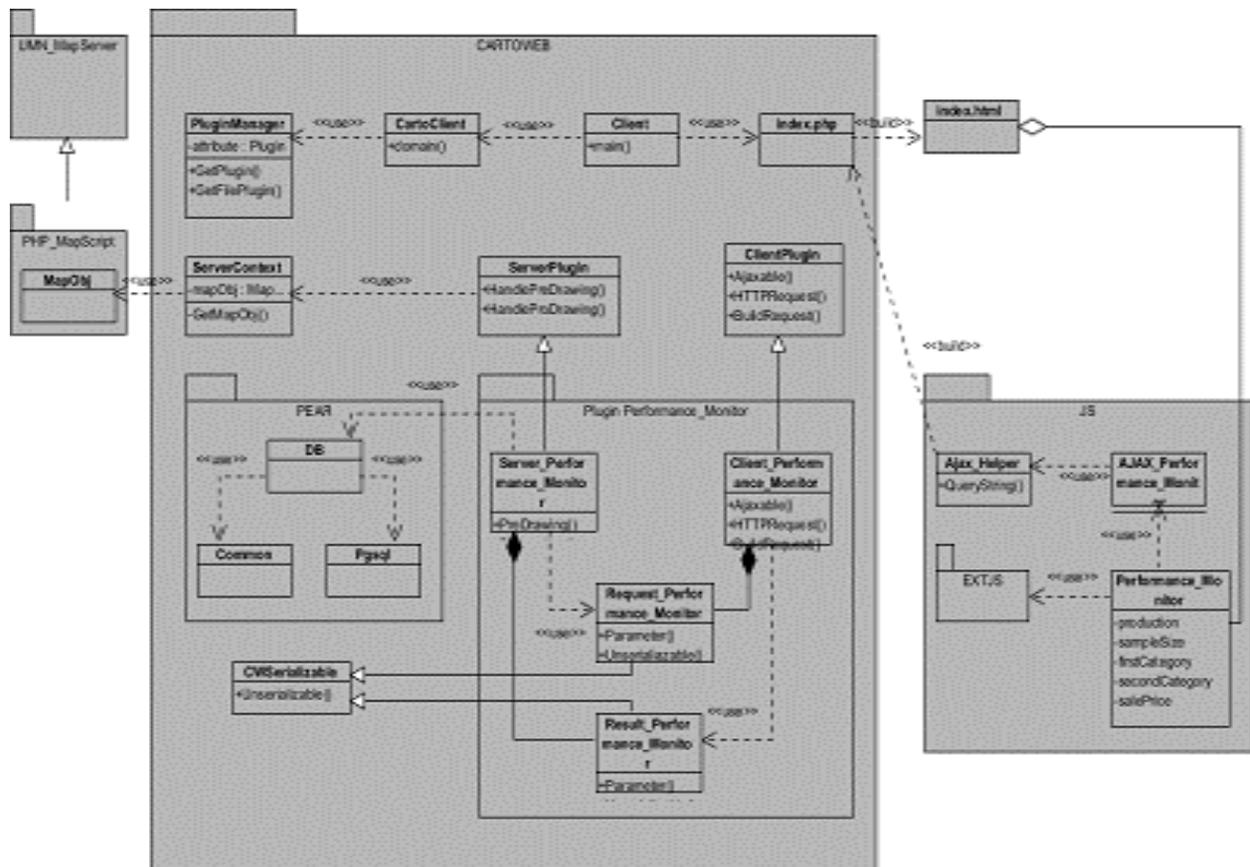


Figura 2 Diagrama de clases del paquete Monitor de rendimiento del módulo de análisis.

3.5.1 Descripción de las clases del diagrama

Client: contiene todos los archivos específicos de PHP del lado de CartoClient y permite la interacción entre la index.php y la CartoClient.

CartoClient: integra y recoge todos los datos y funciones realizadas por cada una de las .js que intervienen en el plugin, y define una serie de variables globales que van a ser utilizadas por la aplicación.

PluginManager: se utiliza para gestionar la base de *plugins*.

ClientPlugin: contiene las interfaces necesarias para los *plugins* del lado del cliente.

ServerPlugin: proporciona la base de herramientas para el desarrollo de *plugins*.

ServerContex: es la contenedora de la información común que ha de ser utilizada por la parte cliente y la servidora, empleando la información seleccionada como un objeto para un fácil manejo de los datos.

MapObj: es donde se definen los métodos, funciones, así como el lenguaje para el intercambio de datos con el servidor de mapas (MapServer).

DB: es la clase encargada de establecer la conexión con el servidor de base de datos para procesar los objetos a editar.

Common: es la encargada de administrar las conexiones a la base de datos para ejecutar las consultas a satisfactoriamente, incluye el tratamiento de los datos.

Pgsql: gestiona desde PHP las funciones de postgreSQL.

CwSerializable: se encarga de serializar todas aquellas clases que pueden ser serializadas, permitiendo la comunicación entre el ClientGZona y ServerGZona.

AJAXHelper: tiene como propósito enviar las respuestas de los *plugins* "AJAX" a los *plugins* que responden a las peticiones del usuario.

index.html: Es la encargada de mostrar en el mapa la información necesaria para la gestión de las zonas de interés.

AJAX_Performance_Monitor: Se encarga de gestionar el pedido y respuesta a las peticiones del usuario por AJAX.

Performance_Monitor: Esta clase es la que muestra la información con el resumen informativo de un ciclo productivo.

Server_Performance_Monitor: Es la clase servidora la cual tiene como principal función la conexión con la base de datos para efectuar los cambios requeridos y enviar las respuestas necesarias al Client_Performance_Monitor.

Client_Performance_Monitor: Recoge y selecciona de las .js contenidas en el paquete JS, toda la información correspondiente a los datos a precisar, entrados a través de los formularios, y los envía al Server_Performance_Monitor.

Request_Performance_Monitor: Esta es una clase común encargada de transportar los datos recogidos en Client_Performance_Monitor desde la interfaz y transportarlos a la clase Server_Performance_Monitor.

Result_Performance_Monitor: Esta también es una clase común, pero se encargada de transportar los datos generados en Server_Performance_Monitor a la clase Client_Performance_Monitor.

3.5.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces. En esencia un patrón de diseño resulta ser una solución a un problema. Para que una solución sea considerada un patrón debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

✓ **Patrones GOF** (Larman, 2003):

Acción: Se utiliza en el proceso de petición de una información cualquiera al sistema por un cliente, mediante la Interfaz Gráfica de Usuario (GUI) (Larman, 2003). Debido a la naturaleza del sistema las interfaces de usuario son fundamentales ya que es la interacción con el usuario una de las características principales que debe poseer. En la clase AjaxPerformanceMonitor, la cual se encarga de manejar las peticiones desde la interfaz de usuario y el servidor, se evidencia este patrón.

Solitario: El patrón solitario se emplea para garantizar que el acceso a una clase sea mediante una única instancia o clase intermedia, lo cual permite controlar el acceso a estas. En el presente caso, con el fin de crear el objeto “mapa” se utiliza la clase ServerContext.

Observador: El patrón observador no es más que una relación lógico-funcional entre varias clases donde una clase observa a *m* clases en busca de cambios en estas con el fin de aplicar un conjunto de acciones. Este patrón se evidencia en el componente Show_Alert, el cual observa los cambios en los cultivos para notificar al usuario de eventos importantes.

✓ **Patrones GRASP** (Larman, 2003):

Experto: Asignar la responsabilidad de realizar una labor a la clase que contiene toda la información necesaria para cumplir la tarea encomendada.

Se aplica, en este caso, en todas las clases de especificidad de las clases Clientplugin y ServerPlugin, las cuales manejan los datos de entrada y los servidores de base de datos y mapas respectivamente. El uso de este patrón permite un alto grado de encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, influyendo a su vez en el bajo acoplamiento del sistema.

Creador: Se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la

información necesaria para ello. Esto es similar al patrón experto, ya que una clase creadora debe ser a su vez experta en el objeto que desea crear. Este patrón se evidencia en las clases anteriormente mencionadas ya que son las encargadas de crear las instancias de las clases Request y Result.

Bajo Acoplamiento: Se asignan las responsabilidades de forma tal que cada clase dependa del menor número de clases posible, minimizando las afectaciones que un cambio en una de ellas acarrearía en las otras.

El acoplamiento es una medida de la fuerza en que una clase está conectada a otras clases, que las conoce y recurre a ellas. En este caso, se refleja el bajo acoplamiento, en cada una de las clases del sistema con el objetivo de que una clase no dependa de muchas clases, de esta forma, no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar. (Universidad de las Ciencias Informáticas)

Alta Cohesión: Asigna a las clases responsabilidades que trabajan sobre una misma área de la aplicación y que no posea mucha responsabilidad, evitando que una clase sea responsable de muchas tareas que puedan dividirse entre otras clases. El uso de este patrón permite que se pueda mejorar la claridad y facilidad en que se entiende el diseño, se simplifique el mantenimiento y existan mejoras de funcionalidad. En las arquitecturas basadas en componente la cohesión es una de las prácticas más importantes ya que permite tener una relación componente-componente satisfactoria y adecuada, en estrecha relación y nivel con el acoplamiento.

Controlador: Un controlador es una interfaz no destinada al usuario que se encarga de manejar un evento o comportamiento del sistema. Define además el modo de operación. En esencia es una clase que funciona como reguladora en un determinado ámbito. Por ejemplo, las clases AJAX_Nombre_Del_Plugin encargada del comportamiento del mismo gestionan todos los eventos que ocurren en este.

3.6 Modelo de despliegue

La vista o modelo de despliegue no es más que una representación abstracta o física, aunque generalmente esta última, que describe cómo se va a encontrar distribuido el sistema, en esta se incluyen interfaces de comunicación y las características de estas (Kruchten, 2004).

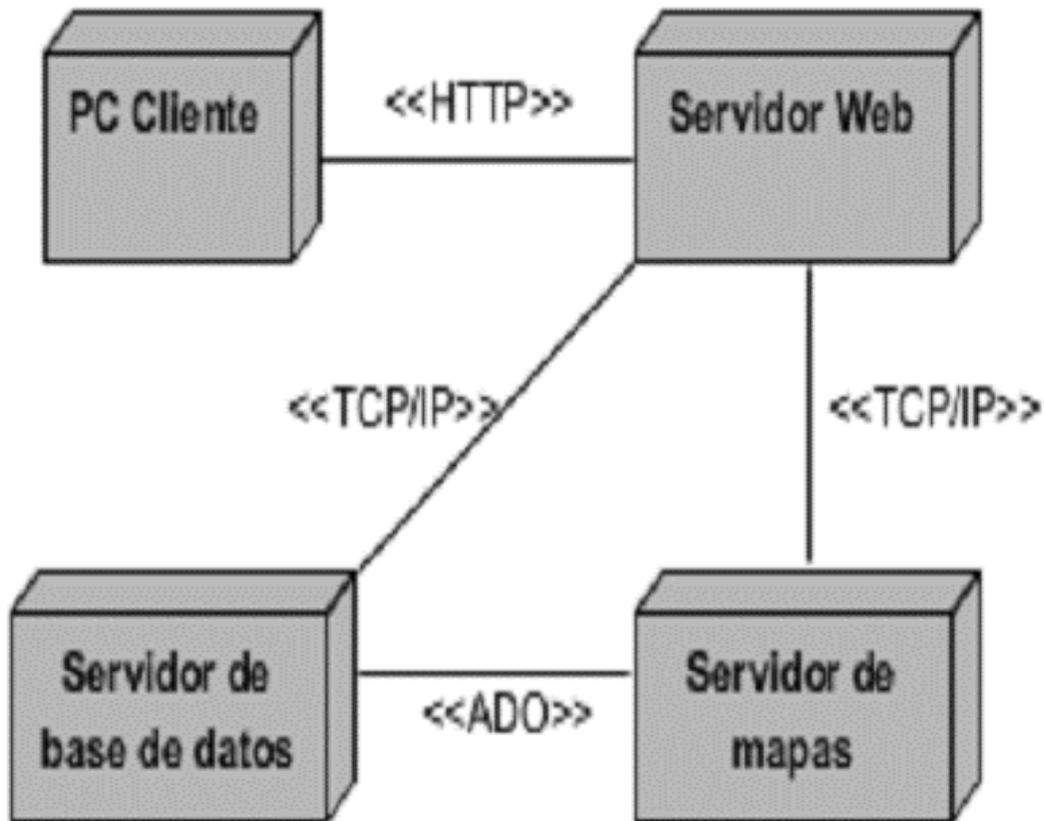


Figura 3 Diagrama de despliegue.

PC Cliente: Representa a los dispositivos que han de conectarse al sistema con el fin de consultar o gestionar información.

Servidor Web: Es el ordenador encargado de recibir las solicitudes del usuario, para lo cual dispone una interfaz web, y en dependencia de esta realizar una solicitud a los servidores de mapa y/o base de datos.

Servidor de base de datos: Posee un servidor de base de datos con los datos asociados tanto a la lógica de negocio como a la información geográfica del sistema.

Servidor de mapas: Es el encargado de manejar la información geográfica de la base de datos así como enviarla para su visualización al servidor web cuando este la requiera.

3.7 Diagrama de componentes

Los diagramas de componentes representan cómo un sistema de *software* es dividido en componentes y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, módulos, ejecutables y paquetes. Estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la

organización y las dependencias entre un conjunto de componentes. En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema (Kruchten, 2004).

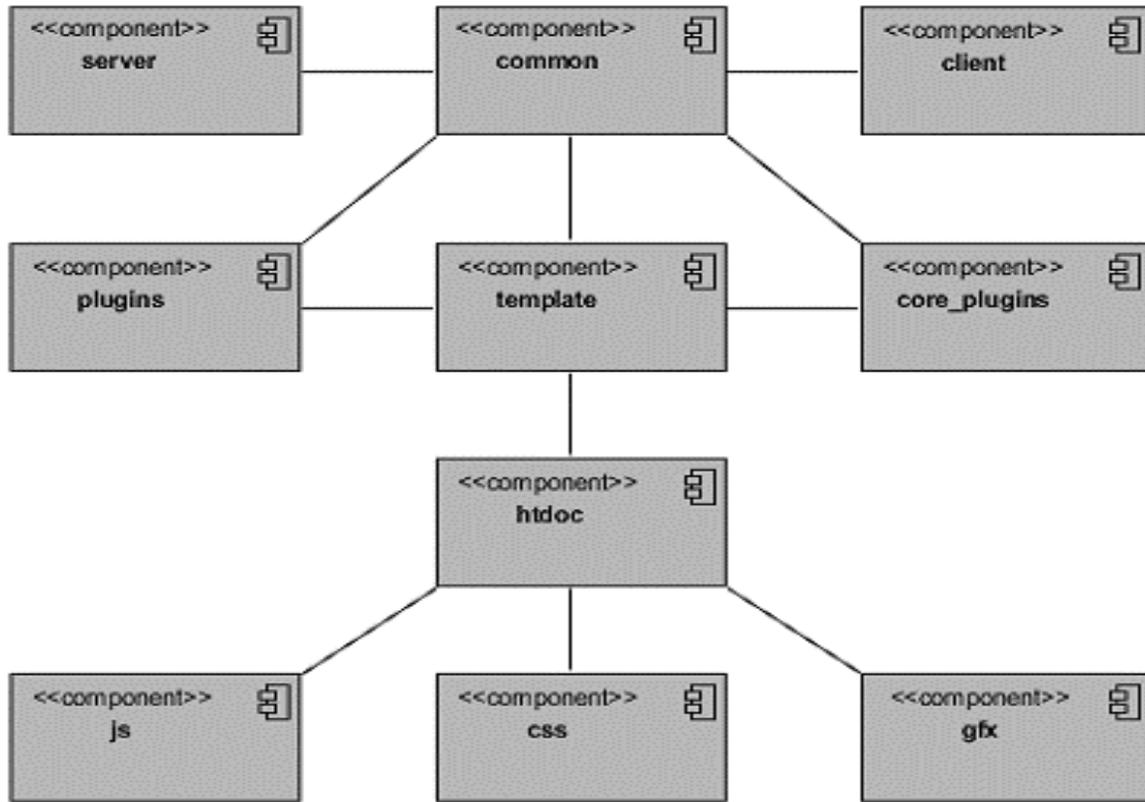


Figura 4 Diagrama de componentes.

Las relaciones fundamentales están dadas entre el paquete *Client*, *Common* y *Server*. En el primero se agrupan los componentes necesarios a ejecutarse en el servidor para que el cliente pueda recibir las respuestas a las peticiones del usuario. En el paquete *Server* se encuentran los componentes propios del servidor que garantizan el funcionamiento del sistema. El paquete *Common* agrupa los componentes comunes tanto para el Cliente como para el Servidor, además garantiza la comunicación entre ellos. En el paquete *Plugins* se encuentran los *plugins* que se van incorporando a la plataforma, mientras que en el paquete *Core-Plugins* están los *plugins* bases, que se inician al ejecutarse el sistema, sin necesidad de ser invocados. El paquete *Template* incluye las variables que garantizan la creación dinámica de páginas. En el paquete *htdoc* se encuentran los proyectos creados, así como archivos *js*, *css* y *gfx*. La relación de este paquete con el resto está dada por el paquete *Template*.

3.8 Pruebas de software

Las pruebas de *software* son investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto. Es una actividad en el proceso de control de calidad. Dependiendo del tipo de pruebas, estas podrán ser implementadas en cualquier momento del proceso de desarrollo. (Universidad de las Ciencias Informáticas)

Así como existen distintos modelos de desarrollo de *software*, existen también modelos de pruebas. Estos modelos definen distintos tipos de pruebas y las agrupan en dos tipos principales, estáticas y dinámicas. Las pruebas estáticas son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación. Pueden referirse a la revisión de documentos, realizar un control de cambios en el flujo de desarrollo entre otras. Las dinámicas en cambio son aquellas que para su ejecución requieren la ejecución de la aplicación o de parte de esta. Permiten el uso de técnicas de caja negra y caja blanca con mayor amplitud. En los subsiguientes epígrafes se hará referencia a dichas técnicas, su implementación y resultados de estas. A continuación, se hace alusión a los niveles a aplicar y una breve descripción de estos.

- **Unitarias:** es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado, una función o un procedimiento, en diseño orientado a objetos, una clase.
- **Sistema:** las pruebas de sistema buscan discrepancias entre el sistema y sus objetivos, es decir, sus requisitos. Estas pruebas verifican que el sistema como un todo responda a las necesidades que le dieron origen y sea así de utilidad al usuario final.

3.9.1 Pruebas de caja blanca

Las pruebas de caja blanca (también conocidas como pruebas estructurales) se centran en los detalles procedimentales del *software*, por lo que su diseño está fuertemente ligado al código fuente del mismo. El diseñador de las pruebas escoge distintos valores de entrada para analizar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados. Las pruebas de caja blanca deben garantizar que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método, ejercitar todas las decisiones lógicas, ejecutar todos los bucles y ejercitar las estructuras internas de datos para asegurar su validez. En esencia estas pruebas se encargan de verificar la sintaxis del código y su funcionamiento lógico, que no existan errores en sus procedimientos y que los datos se manejen correctamente (Pressman, y otros, 2010). Como detalle importante debe hacerse notar que, al estar fuertemente ligadas al código, los cambios en el mismo acarrearán un rediseño de las pruebas. La técnica del camino básico permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico de casos de prueba, este conjunto se

aplica después sobre el código para asegurar que al menos una vez se verifique cada sentencia del mismo. No obstante, se opta por la herramienta PHPUnit como herramienta de automatización de pruebas unitarias con el fin de automatizarlas y solo se aplica la técnica del camino básico en algunas unidades de código de pequeño tamaño. Se ha decidido aplicar pruebas unitarias y de sistema. A continuación, una breve explicación.

La aplicación de estas pruebas en el sistema fue imperante en cada uno de los *plugins* implementados por el equipo de desarrollo. Las pruebas fueron realizadas con la herramienta PHPUnit y pares de entrada-salida hasta eliminar los errores detectados no necesitándose de más de dos iteraciones en ningún caso. Si bien la no detección de más errores no significa su ausencia se ha decidido detener las pruebas con los resultados actuales debido al cumplimiento de los requisitos de calidad.

3.9.2 Pruebas de caja negra

Las pruebas de caja negra se centran principalmente en lo que se quiere de un módulo o sección específica de un *software*, es decir, de su comportamiento deseado. Las pruebas de caja negra son pruebas funcionales, lo que quiere decir que se centran en las funcionalidades del sistema y se diseñan con estas en mente. Estas pruebas deben permitir detectar funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos, errores de rendimiento y errores de inicialización y terminación (Pressman, y otros, 2010).

Entre las técnicas de caja negra más utilizadas se encuentra la de partición de equivalencia. Esta técnica permite dividir las entradas en clases que permiten verificar varias funcionalidades, la capacidad de clasificar permite establecer valores de prueba que determinen datos válidos e inválidos entre otras posibles clasificaciones. Esta técnica se ha empleado en la validación de interfaces y para las pruebas de aceptación se han definido métricas de calidad y creado listas de chequeo que las verifiquen.

3.9 Diseño de los casos de prueba

En el siguiente epígrafe se abordarán los elementos del diseño de las pruebas a aplicar en el sistema, siendo estos elementos: los escenarios, clases de equivalencia y casos de pruebas. Además, solo se documentan las pruebas de interfaz funcionales a nivel de sistema realizadas. Las pruebas de unidad se han realizado mediante el empleo de la herramienta PHPUnit y se ha decidido que el equipo de desarrollo no cuenta con un ambiente de pruebas óptimo para realizar las pruebas a los requisitos no funcionales de hardware y de eficiencia, por ello se recomienda la aplicación de estas en un ambiente controlado y por una entidad certificadora ajena al equipo de trabajo. Por motivos de espacio y organización solo se hace alusión

a los componentes de diseño de dos casos de prueba para el requisito funcional RF 37: Monitor de rendimiento.

Para más información en el diseño de los casos de prueba, dirigirse a los documentos anexos de pruebas y al documento: "Pruebas. Diseño y resultados.docx", para una mayor comprensión de los ejemplos que aquí se exponen.

Tabla 5: Clases de equivalencia.

Clase	Datos	Objetivo
37C	Todos los datos son correctos.	Verificar cumplimiento del requisito funcional numero 18: Generar informe de rendimiento, mediante la técnica de caja negra para el resultado esperado: correcto. Garantizando que el requisito ha sido correctamente implementado y detectar deficiencias a la interfaz.
37I-1	El valor de producción no es numérico, los demás valores son correctos.	Verificar cumplimiento de las validaciones pertinentes al requisito funcional numero 18: Generar informe de rendimiento, mediante la técnica de caja negra para el resultado esperado: incorrecto. Garantizando que la interfaz de entrada de datos para el análisis de rendimiento valida sus campos correctamente, muestra los errores esperados y detectar deficiencias a la interfaz.

Tabla 6: Diseño del caso de prueba al RF 37, método caja negra, resultado esperado: correcto. Clase de equivalencia 18C.

Id:	Módulo:	Resultado esperado	Resultado obtenido
RF37_CN-C	Análisis	Correcto	
		Fecha de ejecución:	Estado:
Responsable:	Enmanuel Adalberto Naranjo Napoles		Pendiente
Pre-condiciones:		Secuencia lógica:	

<ol style="list-style-type: none"> 1. Debe existir una alerta de recolección en la interfaz. 2. El usuario selecciona la opción generar reporte de rendimiento. 3. Han de insertarse los datos pertinentes en la nueva interfaz. 	<ol style="list-style-type: none"> 1. El usuario selecciona la opción generar reporte de rendimiento y presiona el botón cosechar de una alerta de recolección. 2. El usuario introduce los datos solicitados. 3. El usuario presiona generar.
Post-condiciones:	
<ol style="list-style-type: none"> 1. La parcela es modificada en la base de datos cambiando los valores de "sembrada" y "cultivo" a "no" y "ninguno" respectivamente. 2. Se genera una interfaz para introducir los datos de producción del reporte de rendimiento. 3. Se muestra el reporte de rendimiento. 	
Datos necesarios:	<ol style="list-style-type: none"> 1. Producción (Formato: Numérico, Obligatorio) 2. Tamaño de la muestra (Formato: Numérico, Obligatorio) 3. Cantidad de productos de 1ra categoría (Formato: Numérico, Obligatorio) 4. Cantidad de productos de 2da categoría (Formato: Numérico, Obligatorio) 5. Precio de venta (Formato: Numérico, Obligatorio)
Clase o conjunto de entrada probado.	Clase de equivalencia 18C.
Observaciones:	
Descripción:	El sistema, dado un conjunto de datos, debe ser capaz de realizar un resumen informativo acerca de la producción, calidad, aprovechamiento y ganancia-perdida en un ciclo productivo. Este resumen debe mostrarse siempre que se coseche un cultivo y el usuario seleccione la opción reporte de rendimiento en la interfaz.

Tabla 7: Prueba al RF 37, método caja negra, resultado esperado: incorrecto. Clase de equivalencia 18I-1.

Id:	Módulo:	Resultado esperado	Resultado obtenido
RF37_CN-I1	Análisis	Incorrecto. Mensaje de error.	

		Fecha de ejecución:	Estado:
Responsable:	Enmanuel Adalberto Naranjo Napoles		Pendiente
Pre-condiciones:	Secuencia lógica:		
<ol style="list-style-type: none"> 1. Debe existir una alerta de recolección en la interfaz. 2. El usuario selecciona la opción generar reporte de rendimiento. 3. Han de insertarse los datos pertinentes en la nueva interfaz. 	<ol style="list-style-type: none"> 1. El usuario selecciona la opción generar reporte de rendimiento y presiona el botón cosechar de una alerta de recolección. 2. El usuario introduce los datos solicitados. 3. El usuario presiona generar. 		
Post-condiciones:			
<ol style="list-style-type: none"> 1. Se muestra un error en la interfaz especificando: "La producción ha de ser un número real". 			
Datos necesarios:	<ol style="list-style-type: none"> 1. Producción (Formato: Numérico, Obligatorio) 2. Tamaño de la muestra (Formato: Numérico, Obligatorio) 3. Cantidad de productos de 1ra categoría (Formato: Numérico, Obligatorio) 4. Cantidad de productos de 2da categoría (Formato: Numérico, Obligatorio) 5. Precio de venta (Formato: Numérico, Obligatorio) 		
Clase o conjunto de entrada probado.	Clase de equivalencia 18_I-1.		
Observaciones:			
Descripción:	El sistema, dado un conjunto de datos, debe ser capaz de realizar un resumen informativo acerca de la producción, calidad, aprovechamiento y ganancia-perdida en un ciclo productivo. Este resumen debe mostrarse siempre que se coseche un cultivo y el usuario seleccione la opción reporte de rendimiento en la interfaz.		

3.10 Resultados de las pruebas

En el presente epígrafe se hace alusión a los resultados de las pruebas de interfaz aplicadas al sistema, para los dos casos de pruebas del epígrafe anterior.

Tabla 8: Prueba al RF 37, método caja negra, resultado esperado: correcto. Clase de equivalencia 18C.

Id:	Módulo:	Resultado esperado	Resultado obtenido
RF37_CN-C	Análisis	Correcto	Correcto
		Fecha de ejecución:	Estado:
Responsable:	Enmanuel Adalberto Naranjo Napoles	16/05/2017	Terminado
Pre-condiciones:		Secuencia lógica:	
<ol style="list-style-type: none"> 1. Debe existir una alerta de recolección en la interfaz. 2. El usuario selecciona la opción generar reporte de rendimiento. 3. Han de insertarse los datos pertinentes en la nueva interfaz. 		<ol style="list-style-type: none"> 1. El usuario selecciona la opción generar reporte de rendimiento y presiona el botón cosechar de una alerta de recolección. 2. El usuario introduce los datos solicitados. 3. El usuario presiona generar. 	
Post-condiciones:			
<ol style="list-style-type: none"> 1. La parcela es modificada en la base de datos cambiando los valores de "sembrada" y "cultivo" a "no" y "ninguno" respectivamente. 2. Se genera una interfaz para introducir los datos de producción del reporte de rendimiento. 3. Se muestra el reporte de rendimiento. 			
Datos necesarios:	<ol style="list-style-type: none"> 1. Producción (Formato: Numérico, Obligatorio) 2. Tamaño de la muestra (Formato: Numérico, Obligatorio) 3. Cantidad de productos de 1ra categoría (Formato: Numérico, Obligatorio) 4. Cantidad de productos de 2da categoría (Formato: Numérico, Obligatorio) 5. Precio de venta (Formato: Numérico, Obligatorio) 		
Clase o conjunto de entrada probado.	Clase de equivalencia 18C.		

Observaciones:	➤ Añadir consejo al cuadro de selección para su comprensión por parte del usuario.
Descripción:	El sistema, dado un conjunto de datos, debe ser capaz de realizar un resumen informativo acerca de la producción, calidad, aprovechamiento y ganancia-perdida en un ciclo productivo. Este resumen debe mostrarse siempre que se coseche un cultivo y el usuario seleccione la opción reporte de rendimiento en la interfaz.

Tabla 9: Prueba al RF 37, método caja negra, resultado esperado: incorrecto. Clase de equivalencia 18I-1.

Id:	Módulo:	Resultado esperado	Resultado obtenido
RF37_CN-I1	Análisis	Incorrecto. Mensaje de error.	Incorrecto. Mensaje de error.
		Fecha de ejecución:	Estado:
Responsable:	Enmanuel Adalberto Naranjo Napoles	16/05/2017	Terminado
Pre-condiciones:		Secuencia lógica:	
<ol style="list-style-type: none"> 1. Debe existir una alerta de recolección en la interfaz. 2. El usuario selecciona la opción generar reporte de rendimiento. 3. Han de insertarse los datos pertinentes en la nueva interfaz. 		<ol style="list-style-type: none"> 1. El usuario selecciona la opción generar reporte de rendimiento y presiona el botón cosechar de una alerta de recolección. 2. El usuario introduce los datos solicitados. 3. El usuario presiona generar. 	
Post-condiciones:			
<ol style="list-style-type: none"> 1. Se muestra un error en la interfaz especificando: "La producción ha de ser un número real". 			
Datos necesarios:	<ol style="list-style-type: none"> 1. Producción (Formato: Numérico, Obligatorio) 2. Tamaño de la muestra (Formato: Numérico, Obligatorio) 3. Cantidad de productos de 1ra categoría (Formato: Numérico, Obligatorio) 		

	<p>4. Cantidad de productos de 2da categoría (Formato: Numérico, Obligatorio)</p> <p>5. Precio de venta (Formato: Numérico, Obligatorio)</p>
Clase o conjunto de entrada probado.	Clase de equivalencia 18_I-1.
Observaciones:	<ul style="list-style-type: none"> ➤ Añadir consejo al cuadro de selección para su comprensión por parte del usuario. ➤ Cambiar el texto del error por: La producción ha de ser un número real.
Descripción:	El sistema, dado un conjunto de datos, debe ser capaz de realizar un resumen informativo acerca de la producción, calidad, aprovechamiento y ganancia-perdida en un ciclo productivo. Este resumen debe mostrarse siempre que se coseche un cultivo y el usuario seleccione la opción reporte de rendimiento en la interfaz.

Tabla 10: Pruebas de aceptación, pruebas alfa, RF 37.

PRUEBAS ALFA			
Encargado:	Enmanuel Adalberto Naranjo Napoles	Fecha:	20-06-2017
Funcionalidad:	Entorno de prueba:	Resultado de Prueba:	Observaciones:
Monitor de rendimiento	Laboratorio de desarrollo	Exitoso	Reorganizar la interfaz con respecto al orden lógico de los campos de entrada.

Tabla 11: Pruebas de aceptación, pruebas alfa, RF 37. Resultados.

Datos iniciales					
Fecha:	20-06-2017	Tipo de Prueba:	Aceptación	Nº de Caso de prueba:	37A
Característica:	Alertas.	Sub-característica:	Monitor de rendimiento.		

Preguntas de la lista de chequeo	
Fiabilidad: ¿Es la información mostrada cierta?	
Fiabilidad: ¿Es la información obtenida de los cultivos correcta?	
Corrección: ¿Muestra el sistema las alertas?	
Corrección: ¿Permite seleccionar la opción Generar reporte de rendimiento?	
Corrección: ¿Muestra la información deseada en el reporte?	
Información del Caso de Prueba	
Descripción:	El caso de prueba verifica que el procedimiento para generar un reporte de rendimiento sea aceptable para el usuario.
Procedimiento del caso de prueba	
1.- Pasos a seguir:	<ol style="list-style-type: none"> 1. El usuario selecciona cosechar de una alerta de cosecha. 2. El usuario introduce los datos pertinentes. 3. El usuario presiona el botón Generar reporte de rendimiento.
2.- Condiciones externas:	<ol style="list-style-type: none"> 1. Existe al menos un cultivo en el sistema. 2. El cultivo se encuentra cerca de su fecha de cosecha. 3. Existe una alerta de cosecha presente en la interfaz de alertas.
Resultados	
Resultados obtenidos:	Caso de prueba
Se aprueba el caso de prueba, el sistema muestra los datos correctamente y con precisión.	Aprobado.
	Severidad de la falla
	No procede
Observaciones	<ul style="list-style-type: none"> ➤ Reorganizar la interfaz con respecto al orden lógico de los campos de entrada. ➤ Cambiar consejo del botón cancelar.

3.11 Resumen de resultados

Durante la ejecución de las pruebas de *software* se detectaron un conjunto de no conformidades, a continuación, se realiza un resumen por iteración de las no conformidades detectadas.

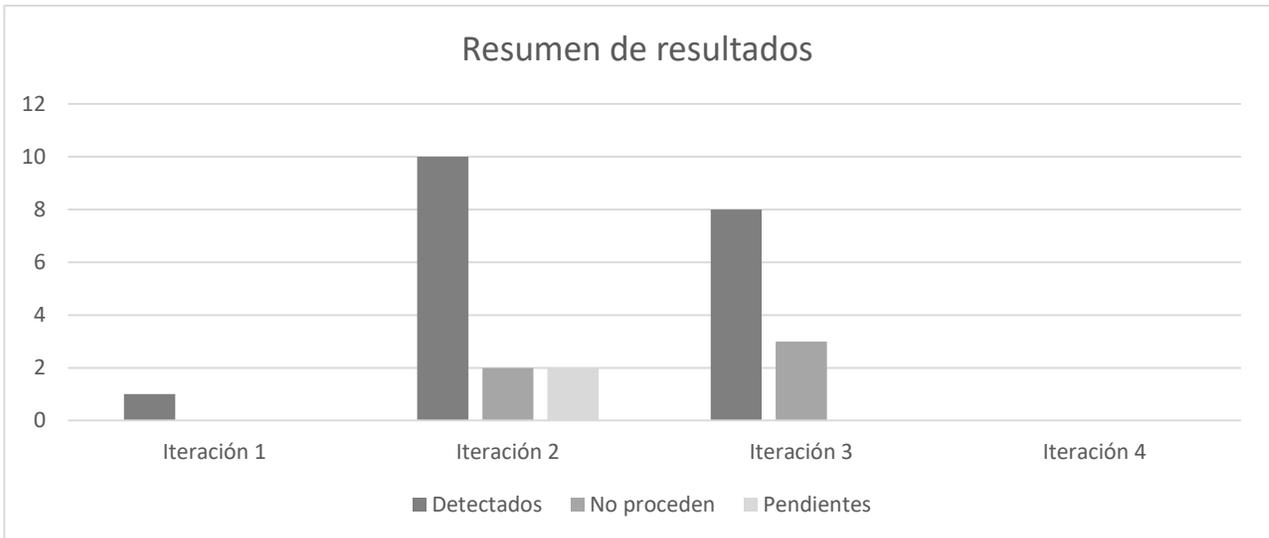


Figura 5 Resultados de las pruebas.

Las pruebas unitarias fueron aplicadas mediante el empleo de la herramienta PHPUnit para automatizar el proceso. A continuación, un resumen de todas las no conformidades detectadas en cada una de las iteraciones:

Primera iteración:

- 1 de análisis.

Segunda iteración:

- 3 de análisis.
- 4 de interfaz
- 3 implementación.

Tercera iteración:

- 1 de análisis.
- 2 de interfaz.
- 2 de implementación.
- 4 funcionales.

3.12 Conclusiones parciales

En este capítulo se exponen los patrones de arquitectura que guiaron el diseño de la aplicación, los cuales se basan fundamentalmente en la estructura del *framework* de desarrollo. Se recogen algunos de los

principales artefactos generados, los cuales permiten definir el sistema de forma clara mediante el empleo del lenguaje de modelado seleccionado. Se tratan los requisitos y se realizan las pruebas, las cuales demuestran que la aplicación responde a las necesidades que dieron inicio al proceso. A partir de estas pruebas se pudo valorar el proceso de desarrollo en términos de calidad y respeto a las etapas y normas de la metodología. Se comprueba que el producto informático da respuesta a las necesidades que le dieron origen mediante la verificación, a nivel de sistema, de cada uno de los requisitos definidos. También se descubrieron las principales debilidades del equipo de desarrollo lo cual permite, en futuras investigaciones prever estas insuficiencias mejorando así los resultados. Se obtiene completamente construido el sistema, el cual permite apoyar los procesos de monitoreo y control de los recursos agrícolas en las entidades agropecuarias cubanas, lo cual proveerá a dichas entidades de un mecanismo de regulación y control informatizado que apoye los procesos de toma de decisiones. Por último, ha sido en este capítulo donde se han adquirido más conocimientos, relacionados con el empleo practico y real de la información geográfica y sus ventajas.

Conclusiones

Luego de haber culminado la investigación se concluye que:

- ✓ La definición de los elementos teóricos aportó la base de la investigación científica, sirvió de guía a los procesos de obtención y análisis de la información. Gracias a esto se pudieron obtener resultados científicos en concordancia con los elementos definidos, campo de acción, objeto de estudio, problema, objetivo general, preguntas científicas y tareas.
- ✓ Existen diversos problemas que afectan la calidad de los servicios y productos brindados por las entidades agrícolas cubanas, estos problemas están estrechamente vinculados entre sí y responden principalmente a un control no eficiente de los recursos y procesos que en ellas se realizan.
- ✓ Las causas asociadas a estos problemas de control son, generalmente, debido a la no centralización de la información y el empleo de técnicas manuales en demasiados procesos, lo cual puede involucrar como factor decisivo el error humano.
- ✓ Estos problemas están presentes debido a la no informatización de los procesos de la entidad, lo cual sucede debido a la no existencia de una solución informática que satisfaga sus necesidades, aunque existen investigaciones y sistemas que abordan parte de la problemática estos fueron sopesados y encontrados deficientes como solución, aunque muy útiles como fuentes de información y conocimientos.
- ✓ La selección de las herramientas y tecnologías es fundamental para la calidad del producto, la selección de estas es la que define el alcance y utilidad que puede poseer el sistema, su seguridad, y muchas otras restricciones. Es por eso que se intentan emplear herramientas de conocida calidad y eficiencia, permitiendo así asegurar, al menos hasta cierto punto, la calidad y estabilidad del *software*.
- ✓ Se ha construido un sistema de información geográfica para apoyar los procesos de monitoreo y control de los recursos y procesos agrícolas de las entidades agropecuarias del país; el mismo permite a sus usuarios poseer la información pertinente a los cultivos, insumos y otros elementos disponible, actualizada y segura, además de brindar apoyo a los procesos de toma de decisiones.
- ✓ Se ha dado respuesta a cada una de las preguntas que surgieron para guiar la investigación, las tareas se realizaron como se habían definido, las etapas de la metodología se han respetado, los artefactos se han generado de acuerdo a lo definido por el expediente de proyecto y la metodología, la aplicación implementa de forma correcta todas las funcionalidades definidas y respeta toda restricción establecida sobre la misma. Por tanto, se puede afirmar que se han cumplido con las normas de calidad tanto del proceso como del producto informático en sí.

- ✓ El proceso de desarrollo ha sido enriquecedor para el equipo de trabajo, se han adquirido conocimientos y experiencia valiosa de trabajar en un proyecto tan importante e interesante. Además, la satisfacción y el placer de cumplir con el deber, como ingenieros en ciencias informáticas, de contribuir a la informatización de los procesos de la nación conlleva un crecimiento profesional de incalculable valor para los involucrados en este proceso.

Recomendaciones

Teniendo en cuenta los resultados alcanzados en la investigación se recomienda:

- ✓ Incorporar un módulo para un análisis más detallado de las plagas.
- ✓ Incorporar soporte para sensores de densidad de grano y humedad.
- ✓ Desplegar el sistema en las entidades agropecuarias del país.

REFERENCIAS BIBLIOGRÁFICAS

IBM. Rational Unified Process. 2011. *IBM. Rational Unified Process.* 2011.

Apache. 2013. Documentación del Servidor Apache. [En línea] 2013. [Citado el: 21 de 04 de 2017.]
<http://httpd.apache.org/docs/2.0/es/>.

Apache Software Foundation. Apache Software Foundation. [En línea] [Citado el: 18 de 03 de 2017.]
<https://www.apache.org/>.

Biblioteca virtual- BR. 2014. Banco de la República(Cartografía). *Banco de la República(Cartografía)*. [En línea] 2 de septiembre de 2014. [Citado el: 05 de 03 de 2017.]
<http://www.banrepcultural.org/blaavirtual/ayudadetareas/geografia/cartografia>.

Bongiovanni, Rodolfo , y otros. 2006. *Agricultura de precisión: integrando conocimientos para una agricultura moderna y sustentable.* 2006.

Bravo, Javier Dominguez. 2012. *Los Sistemas de Información Geográfica y las Energías Renovables.* 2012.

Carrasco, Aida Socarrás. 2013. *Sistema de Información Geográfica para el Grupo Estatal AZCUBA.* 2013.

Castañeda Rodríguez, Siomelis. 2011. *Sistema de Información Geográfica para Campismo Popular.* La Habana : s.n., 2011.

Community, NetBeans. Portal del IDE NetBeans. [En línea] [Citado el: 14 de 03 de 2017.]
https://netbeans.org/index_es.html.

Corporation, TEK. Geoagro GIS. *Geoagro GIS.* [En línea] [Citado el: 22 de 11 de 2016.]
<http://site.geoagro.com/es/producto/geoagro-gis>.

Dalia Moreno, Édgar Quiñones, Luis Carlos Tovar. 2014. *Los Sistemas de Alerta Temprana, SAT, una herramienta para la prevención de desastres por inundación y efectos del cambio climático.* s.l. : Revista Ciencias e Ingeniería al Día, 2014.

Díaz, Luis Carlos. Pontificia Universidad Javeriana. [En línea] [Citado el: 18 de 02 de 2017.]
<https://sophia.javeriana.edu.co/~lcdiaz/ADOO2007-3/PatronesGoF-ACCR.pdf>.

Emiliano García, Fernando Flego. 2008. *Agricultura de precisión.* 2008.

Eólica, Asociacion Empresarial. 2008. Asociacion Empresarial Eólica. [En línea] 2008. [Citado el: 12 de 10 de 2016.]
<https://www.aeeolica.org/>.

Galeano Gil, Germán , Díaz Márquez, Pablo y Sánchez Alonso, José Carlos. 2016. *Manual de HTML.* Barcelona : s.n., 2016.

Gámez, Maria Rodriguez. 2012. *LOS SIG Y LA INTEGRACIÓN IBEROAMERICANA A PARTIR DEL USO DE LAS FUENTES RENOVABLES DE ENERGÍA.* 2012.

Garzías, Javier. JavierGarzas. [En línea] [Citado el: 12 de 01 de 2017.] <http://www.javiergarzas.com/2014/08/los-patrones-grasp.html>.

Gómez, Yulaini Ramírez. 2011. *Tesis: Desarrollo de un Sistema de Información Geográfica de recursos hidráulicos sobre protocolo WAP*. 2011.

2012. Gran Turismo. *Gran Turismo*. [En línea] 2012. [Citado el: 12 de 04 de 2017.] <http://www.grntur.blogspot.com/>.

Group, Global Development. 2008. Global Development Group. [En línea] 2008. [Citado el: 18 de 02 de 2017.] <http://www.postgresql.org/files/documentation/pdf/8.3/postgresql-8.3-A4.pdf>.

Grupo de Trabajo AENOR AEN/CTN71/SC7/GT26. 2007. Pruebas de Software ISO/IEC/IEEE 29119. *Pruebas de Software*. 2007.

Gutiérrez Puebla, Javier y Gould, Michael. 1994. *SIG: SISTEMAS DE INFORMACIÓN GEOGRÁFICA*. 1994.

I. Jacobson, G. Bosh. 2000. *El proceso Unificado de Desarrollo de Software*. Madrid : PEARSON EDUCACION.SA, 2000. 84-7829-036-2.

Informáticas, Universidad de las Ciencias. Mejora de Procesos de Software. [En línea] [Citado el: 06 de 02 de 2017.] <http://mejoras.prod.uci.cu/>.

JACOBSON, I. y BOOCH, G. 2000. *El Proceso Unificado de Desarrollo de Software*. 2000.

Jacobson, Ivar, Booch, Grady y Rumbaugh, Jame. 2004. *El Proceso Unificado de Desarrollo. Vol I, Vol II*. La Habana : Felix Varela, 2004.

Jaime M. Tan Nozawa. PHP Experto. [En línea] [Citado el: 14 de 12 de 2016.] <http://phpexperto.blogspot.com/2007/11/php-mapscrip-de-mapserver-parte-1.html>.

Jose Luis Almazán Gárate, M. Carmen Palomino, y Hilda Araceli Caba. 2009. *Sistema de Información Geográfica en la Gestión Integral del Litoral*. Madrid : s.n., 2009.

Larman, Craig. 2003. *UML y Patrones 2da Edición*. España : Pearson Educación : s.n., 2003.

López, Gerardo Sarabia. 2008. *Búsqueda y ponderación de información contenida en Bases de datos Espaciales*. México : Centro de Investigación en computación, 2008.

Manual de HTML. [En línea] [Citado el: 2017 de 04 de 23.] <http://www.desarrolloweb.com/manuales/21/>.

MappingGIS. MappingGIS. [En línea] [Citado el: 14 de 05 de 2017.] <http://mappinggis.com/2012/09/por-que-utilizar-postgis/>.

Martínez, Luis F. Iribarne. 2003. *Modelo de Mediación para el Desarrollo de Software basado en Componentes COTS*. s.l. : Universidad de Almería, 2003.

Medina, A. 2015. Metodología de desarrollo de software. [En línea] 2015. [Citado el: 18 de 02 de 2017.] http://www.academia.edu/4984909/Metodologia_de_desarrollo_de_software.

Membrides Espinosa, Antonio , y otros. 2012. *Desarrollo de Sistemas de Información Geográficos sobre una plataforma soberana*. 2012.

Mirabal, Tatiana Garcia. 2012. *Método para calcular la complejidad de los requisitos funcionales de los proyectos del Centro de Informatización de la Gestión de Entidades.* 2012.

Mozilla Developer Network. MDN. [En línea] [Citado el: 11 de 01 de 2017.]
<https://developer.mozilla.org/es/docs/JavaScript>.

Mozilla. JavaScript|MDN. [En línea] [Citado el: 07 de 04 de 2017.]
<https://developer.mozilla.org/es/docs/Web/JavaScript>.

Olaya, Víctor. 2011. *Sistemas de Información Geográfica.* [Documento] 2011.

Open Source Geospatial Foundation. MapServer. [En línea] [Citado el: 21 de 05 de 2017.] <http://mapserver.org/>.

PEÑA VERA, Tania y PIRELA MORILLO, Johann. 2007. *La complejidad del análisis documental.* s.l. : Información, cultura y sociedad, 2007.

Philippe Rigaux, Michel Scholl, Agnes Voisard. 2002. *Spacial Databases with Application to GIS.* 2002.

PHP. PHP. [En línea] [Citado el: 12 de 12 de 2016.] <http://php.net/manual/es/intro-whatis.php>.

PostgreSQL. PostgreSQL-es. [En línea] [Citado el: 07 de 12 de 2017.]
http://www.postgresql.org/es/sobre_postgresql.

Pressman, Roger S. 2010. *Ingeniería de software. Un enfoque práctico. 7ma Edición.* 2010.

Programa corporativo para el desarrollo tecnológico agroalimentario y agroindustrial del cono sur. 2014. *Manual de agricultura de precisión.* 2014.

Puente, Wilson. 2011. Técnicas de investigación. [En línea] 2011. [Citado el: 18 de 02 de 2017.]
<http://www.rppnet.com.ar/tecnicasdeinvestigacion.htm>.

RAE. Diccionario de la Real Academia de la Lengua Española.

Real Academia Española. 2017. Real Academia Española. [En línea] 2017. <http://rae.es/recursos/diccionarios/drae>.

Roca Díaz., Yanet y Pérez González, Miguel Yoandy. 2012. *Sistema de Información Geográfica para el transporte obrero de la Universidad de las Ciencias Informáticas. Versión 2.0.* [Documento] La Habana : s.n., 2012.

Sánchez, Tamara Rodríguez. 2014. *Metodología de desarrollo para la actividad productiva de la UCI.* 2014.

Scheltón Lima, Annabell, Ramírez Leal, Ramdy y López Cardoso, Osmani. *Tecnologías y herramientas libres para el desarrollo de SIG en la web.*

Sencha. Sencha. [En línea] [Citado el: 27 de 03 de 2017.] <https://www.sencha.com/>.

services, Centers for medicare & medicaid. 2005. *SELECTING A DEVELOPMENT APPROACH.* 2005.

Sphinx. 2014. MapServer. [En línea] 2014. [Citado el: 17 de Enero de 2014.]
<http://www.mapserver.org/es/about.html#about>.

Universidad de las Ciencias Informáticas. *Manual de estudios de ingeniería de software para la carrera de ciencias informáticas.* Habana, Cuba : s.n.

Varen Caballero, Eliani. 2010. *Estrategia para la implementación de Sistemas de Información Geográfica del petróleo sobre la base de la Plataforma GeneSIG. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.* La Habana : s.n., 2010.

Vázquez Manso, Ruben. 2011. *Desarrollo de un Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas GvSIG Mobile.* Ciudad de La Habana : s.n., 2011.

VisualParadigm. VisualParadigm. [En línea] [Citado el: 05 de 02 de 2017.] <http://www.visual-paradigm.com/product/vpuml/>.

W3C. W3C: Ajax introduction. [En línea] [Citado el: 18 de 04 de 2017.] https://www.w3schools.com/xml/ajax_intro.asp.

Zayas Agüero, Pedro Manuel. 2014. *Métodos del conocimiento teórico.* [En línea] 2014. [Citado el: 18 de 02 de 2017.] <http://www.rppnet.com.ar/tecnicasdeinvestigacion.htm>.

Anexos

Anexo 1: Entrevista realizada al MsC. Yuniel Eliades Proenza Arias y un breve resumen de las respuestas.

1. ¿Qué procesos se llevan a cabo?
Siembra, riego, fertilización, fumigación, cosecha.
2. ¿Tienen algún orden o secuencia los mismos?
Siembra, riego, fertilización, fumigación si la cosecha presenta plagas y recolección.
3. ¿Podría explicarme en qué consiste cada proceso?
En esencia es como se explica anteriormente, claro cada proceso es un poco más complejo que eso, por ejemplo, sembrar tiene en cuenta la estación del año y en ocasiones deben de cambiarse los planes de producción para cumplir con la demanda, el riego consiste en regar los cultivos, por supuesto si un proceso se ve afectado de alguna manera todos los que le sigan verán algún tipo de afectación ya que estos se ejecutan consecutivamente. Fertilización se realiza dependiendo del fertilizante, puede ser por aviación personal de a pie, vehículos o incluso mezclándolo con el agua, generalmente se fertiliza antes, durante y después de cultivar una parcela. Fumigación es según sea necesario, también puede realizarse de varias formas, aunque la aviación es una de las más empleadas, también se puede aplicar durante el riego o preventivamente en algunos casos. Cosechar implica maquinaria y mano de obra en el terreno, ellos recolectan los productos y se almacenan o distribuyen según la necesidad.
4. ¿Qué procesos son los más importantes?
Todos, ya que todos responden al ciclo de vida del producto, por tanto, todos son igual de importantes.
5. ¿Qué datos maneja en cada uno de ellos?
En la siembra tenemos en cuenta el terreno, o sea, dimensiones y el tipo de terreno y las semillas; durante el riego se consideran varios factores, en dependencia de la sequía o las precipitaciones se determina la cantidad de agua a emplear, también se intenta que esta no afecte la fertilización ya que algunos fertilizantes ven su efectividad afectada por la humedad del suelo y por supuesto el regadío de los cultivos no es uniforme, sino que varía de un cultivo a otro. La fertilización se realiza de manera planificada según la disponibilidad que se tenga y que sirva para el cultivo claro, aunque se prefieren ciertos fertilizantes para ciertos cultivos ya que son más eficaces que otros, todo depende de la disponibilidad. Fumigación es bastante simple, se emplea el plaguicida asociado a la plaga en cuestión, sino se tiene hay que adquirir alguno que sea aplicable. Durante la cosecha se procede a la recolección de los productos, en dependencia del mismo esta puede realizarse un poco antes o después de la fecha planificada con el objetivo de aumentar su calidad, pero de forma general se intenta cumplir con el límite establecido y con el tiempo de siembra del cultivo evidentemente. Cabe notar que el factor que más influye en todo el proceso es el clima, el cual puede atrasar o acelerar cualquiera de estas etapas e incluso evitarlas.
6. ¿Estos datos como los obtiene?
Los datos los dan los especialistas en su mayoría, pero algunos datos ya se conocen en la entidad, por ejemplo las áreas de cultivo se encuentran asignadas y son de conocimiento de la dirección algunos elementos como lo son la eficacia de los fertilizantes se conocen por la propia experiencia de los campesinos; por ejemplo los plaguicidas pueden ser peligrosos al medio ambiente y el terreno si no se emplean adecuadamente, esta información se obtiene de la información del producto o de especialistas en ellos. El clima mediante informes meteorológicos y el resto mediante la observación de los cultivos.

7. ¿Existen restricciones sobre los datos que se manejan?
Los cultivos son cultivos reales, se hace referencia a la humedad en porciento, las semillas se tratan en unidades, el terreno en caballerías, hectáreas o metros cuadrados; el fertilizante en unidades en dependencia si es liquido o sólido, igual con los plaguicidas.
8. ¿Existen restricciones de seguridad?
Sí, los campesinos se encargan del trabajo manual en el campo y la observación, pero por ejemplo solo los especialistas y jefes de brigada pueden solicitar el empleo de insumos o disponer medios de transporte. El jefe de la entidad es el encargado de aprobar los procesos que impliquen contratar o comprar algún producto o mano de obra. Sin embargo, el acceso al almacén solo está permitido al almacenero, especialistas y a la administración, el primero vela por la integridad y el buen estado del almacén mientras que la administración y los especialistas pueden realizar auditorías o disponer de recursos que se encuentren almacenados ahí. En el caso de los especialistas deben disponer a través de una solicitud mientras que la administración lo realiza de una manera directa.
9. ¿Quiénes deben acceder a la información de la entidad?
Los involucrados en trabajar con la información son los especialista y jefes de brigada, aunque la administración tiene acceso a esta siempre que lo decida pertinente. En muchas ocasiones los campesinos también tienen acceso a esta información, aunque no lo necesitan para desarrollar sus tareas.
10. ¿Quiénes pueden alterar esta información?
Solo la administración y algunos especialistas designados al manejo de los fertilizantes y otros recursos.
11. ¿Existen restricciones de diseño?
Que sea tan fácil o sencillo como sea posible ya que muchos de los trabajadores no poseen un alto grado de escolaridad.
12. ¿Qué diferencias hay en los niveles de acceso?
Estas diferencias se notan principalmente a la hora de consumir recursos, ya que el acceso a la información es bastante abierto pero la alteración de esta o la solicitud de algún insumo debe hacerse a través de mecanismos de control, generalmente de la administración.
13. ¿Hay alguna otra operación que usted realice además de las ya mencionadas que le sirvan de apoyo?
Sí, el proceso de cierre, ahí se analiza el plan de producción y el presupuesto entre otros elementos, además debe de saberse la rentabilidad y el aprovechamiento de los cultivos no críticos, o sea, aquellos que no son parte de la base agrícola y que se producen con otros fines. Y de forma general se hace un resumen del rendimiento, las fechas de liquidación entre otros aspectos relacionados a los resultados.
14. ¿Algún dato importante que desee añadir?
Aunque los procesos no presentan tantos problemas a la hora de ejecutarse si los presentan a la hora de planificarse, los especialistas generalmente utilizan como principal herramienta de obtención de información a los campesinos y la observación que ellos mismos realizan sobre las tierras de cultivo. Aunque siempre contamos con personal experimentado, no todos tienen el mismo punto de vista ni la misma información. Es una necesidad que tenemos, que de alguna manera la información disponible fuera precisa y concreta intentando eliminar el error humano y la demora todo lo posible. Además, un error en una

solicitud de fumigación, puede por ejemplo atrasar el proceso, y como mencionamos anteriormente el atraso en un proceso implica afectaciones a los demás.

Anexo 2: Listado de requisitos

Nº	Nombre	Descripción
RF1	Acercar una región del mapa.	Permitirá aumentar el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de acercar.
RF2	Alejar una región del mapa.	Permitirá disminuir el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de alejar.
RF3	Visualizar todo el mapa.	Permitirá visualizar el mapa según la vista inicial de la aplicación.
RF4	Modificar el centro del mapa.	Permitirá modificar el centro del mapa, sin cambiar la escala del mismo, poniendo como centro el nuevo punto seleccionado por el usuario.
RF5	Mover el mapa.	Permitirá al usuario mover el mapa para visualizar otras regiones.
RF6	Navegar utilizando el mapa de referencia.	Permitirá al usuario visualizar otras regiones en el mapa principal, haciendo uso de un Mapa de Referencia, para posicionarse en el lugar indicado en este último.
RF7	Realizar identificación puntual.	Permitirá al usuario consultar la información asociada a los objetos del mapa. Para ello deberá marcar el punto sobre el mapa y se le mostrará la información de los elementos encontrados en dicho punto.
RF8	Realizar identificación rectangular.	Permitirá al usuario consultar la información asociada a los objetos del mapa que se encuentren dentro de un rectángulo dibujado por este.
RF9	Realizar identificación poligonal.	Permitirá al usuario consultar la información asociada a los objetos del mapa que se encuentren dentro de un polígono dibujado por este.
RF10	Localizar por elemento.	Permitirá al usuario localizar sobre el mapa el elemento deseado. Como resultado se mostrará el mapa de forma tal que dicho elemento quede en el centro.
RF11	Habilitar capas del mapa.	Permitirá al usuario habilitar las capas del mapa que se encuentren deshabilitadas.
RF12	Deshabilitar capas	Permitirá al usuario deshabilitar las capas del mapa que se

	del mapa.	encuentren habilitadas.
RF13	Modificar escala.	Permitirá al usuario cambiar la escala con la que se visualiza el mapa en la aplicación.
RF14	Exportar mapa como imagen.	Permitirá al usuario guardar el mapa mostrado en el sistema como una imagen en formato jpg.
RF15	Realizar tematización por colores.	Permitirá al usuario realizar un análisis sobre el mapa, observando los elementos dibujados según los colores establecidos. Los criterios de análisis son: <ol style="list-style-type: none"> 1. Nivel de humedad. 2. Nivel de fertilización. 3. Cultivos plagados. 4. Estado general del cultivo.
RF16	Medir distancia.	Permitirá medir la distancia entre dos o más puntos especificados sobre el mapa.
RF17	Calcular área y perímetro.	Permitirá calcular el área y perímetro de un polígono dibujado sobre el mapa por el usuario.
RF18	Autenticar usuario.	Permitirá al usuario autenticarse para iniciar su sesión en el sistema.
RF19	Cerrar sesión.	Permitirá al usuario cerrar su sesión en el sistema.
RF20	Editar perfil.	Permitirá al usuario cambiar sus datos.
RF21	Adicionar usuario.	Permitirá adicionar nuevos usuarios para que puedan trabajar con el sistema.
RF22	Eliminar usuario.	Permitirá eliminar los usuarios que trabajan con el sistema.
RF23	Modificar usuario.	Permitirá modificar los datos de los usuarios que trabajan con el sistema.
RF24	Limpiar campos.	Permitirá quitar la información que se muestra en los campos de la interfaz.
RF25	Insertar parcela	Permitirá adicionar parcelas en el sistema.
RF26	Eliminar parcela.	Permitirá eliminar las parcelas en el sistema.
RF27	Modificar parcela.	Permitirá modificar los datos de las parcelas en el sistema.
RF28	Adicionar cultivo.	Permitirá adicionar las cultivos en el sistema.
RF29	Eliminar cultivo.	Permitirá eliminar las cultivos en el sistema.
RF30	Modificar cultivo.	Permitirá modificar los datos de las cultivos en el sistema.
RF31	Adicionar fertilizante.	Permitirá adicionar fertilizantes en el sistema.

RF32	Eliminar fertilizante.	Permitirá eliminar las fertilizantes en el sistema.
RF33	Modificar fertilizante.	Permitirá modificar los datos de los fertilizantes en el sistema.
RF34	Adicionar plaga.	Permitirá adicionar plagas en el sistema.
RF35	Eliminar plaga.	Permitirá eliminar las plagas en el sistema.
RF36	Modificar plaga.	Permitirá modificar los datos de las plagas en el sistema.
RF37	Monitor de rendimiento	El sistema brinda al usuario un resumen informativo con aspectos relacionados al rendimiento de los cultivos.
RF38	Alertas	El sistema debe mostrar alertas cuando se acerca una fecha importante (regadío, fertilización, recolección).