

Universidad de las Ciencias Informáticas

Facultad 6



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

Título: Sistema de Información Geográfica para la Gestión de Cultivos en Cuba

Autores: Dunia Morales Broche

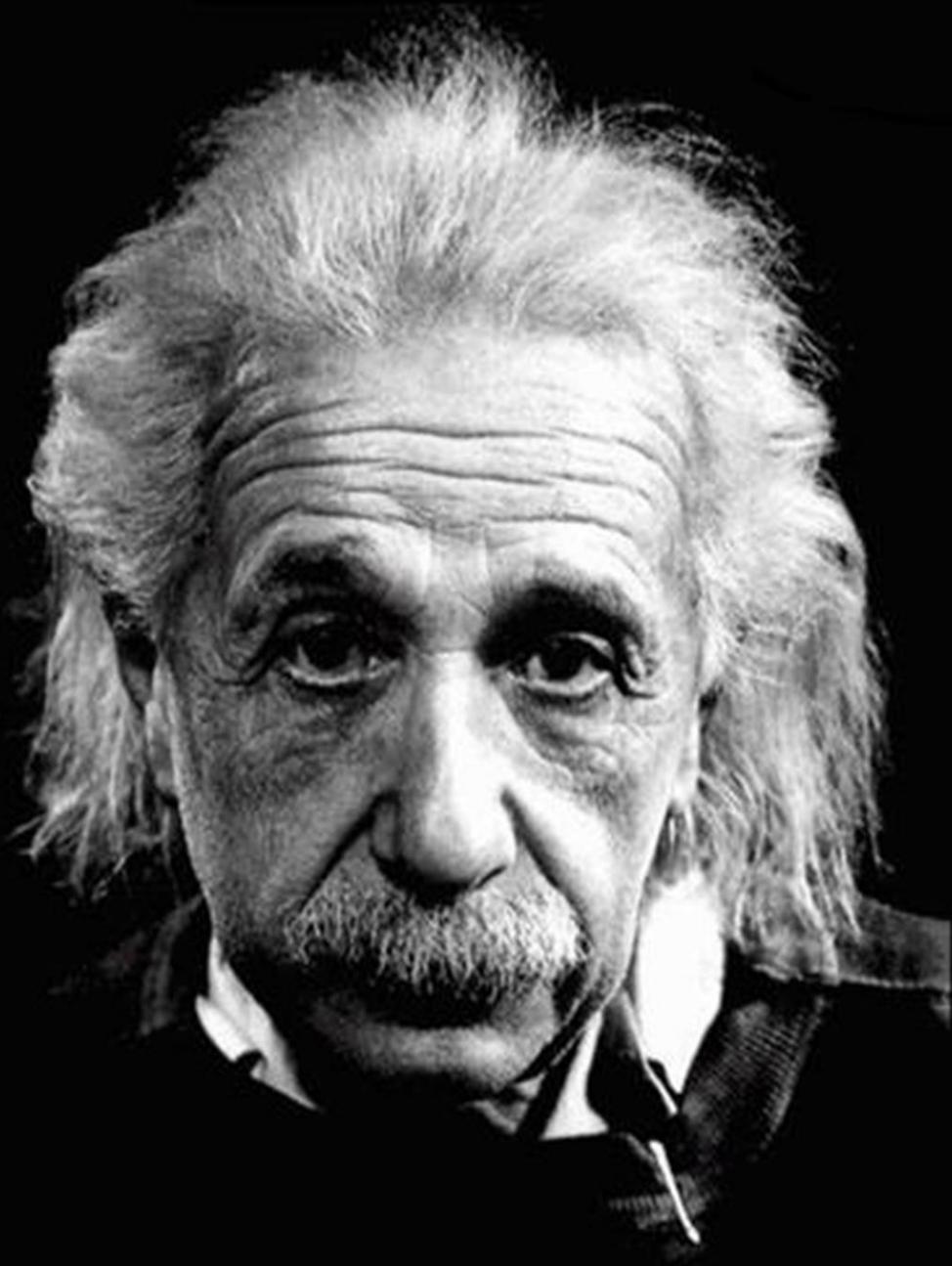
Yeline Elizabeth González Álvarez

Tutor: MsC. Yuniel Eliades Proenza Arias

Co-tutor: Ing. Yoan Mandina Verdecia

La Habana, 2016

“Año 58 de la Revolución”



" No pretendamos que las cosas cambien si seguimos haciendo lo mismo. "

Albert Einstein

Declaración de autoría

Declaro por este medio que nosotras, Yeline Elizabeth González Álvarez y Dunia Morales Broche, con números de carnet de identidad 92030241791 y 91102235378 respectivamente, somos los autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste, firmamos la presente declaración a los días ____ del mes de ____ del año 2016.

Yeline Elizabeth González Álvarez
Autor

Dunia Morales Broche
Autor

MsC. Yuniel Eliades Proenza Arias
Tutor

Ing. Yoan Mandina Verdecia
Co-tutor

Dedicatoria

Dania:

Dedico esta tesis a mi hermana Dania porque se lo orgullosa que se siente de mí en estos momentos y sé que lo disfruta como si fuera ella. Espero que vivas pronto una experiencia igual a la mía, porque nadie mejor que tú lo merece.

A mis padres por guiarme hasta aquí, por ser el mejor ejemplo y dedicarme la vida para sacar lo mejor de mí.

A mi novio por estos años a mi lado, dándome todo el apoyo que he necesitado para poder superar los contratiempos y la distancia de la familia.

Yeline:

Les dedico esta tesis principalmente a mi mamá y a mi palito Ricardo, a quienes con este logro quiero devolver un poco de lo que me han dado, ustedes son la razón de mi vida. Siempre han creído en mí y me han sacado adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Hoy me siento orgullosa de recompensarlos de esta manera, por todo lo que han hecho para que este día llegara.

A mi hermanitos Ariennis y Richar, por estar conmigo y apoyarme siempre. Por compartir sus vidas conmigo, por ser los mejores hermanitos del mundo, los amo. Me duele mucho que no estén presentes en este momento tan importante en mi vida.

Agradecimientos

Dunia:

Primero que nada, quiero agradecer a mi hermana Dania quien es merecedora de este resultado tanto como yo, por compartir y vivir conmigo cada minuto desde que nos crearon en el vientre de mamá. Por no perderse detalle de mí aunque estemos lejos. Por ser mi inspiración con ese espíritu de vencedora y luchadora que le abre caminos donde quiera que esté. Hoy le agradezco a la naturaleza por hacernos juntas creando necesidad enorme de estar ceca que me hizo venir tras sus pasos.

A mis padres Mirayda y Exiquio por todo ese amor, sacrificio, y dedicación que me han brindado desde el día en que nací, por esa virtud de guiarme por el camino correcto sin imposiciones, dejándome ser yo en todo momento y enseñándome a enfrentar obstáculos de manera independiente. Ser hija de ustedes es la mayor bendición que me ha dado la vida.

A mi novio Randy por tanto amor, por estar siempre a mi lado, por ayudarme en este largo recorrido, por ser la razón que me dio fuerzas para continuar cuando todo parecía perdido. Por enseñarme a amar independientemente de los defectos que podamos tener y por construir conmigo una hermosa familia que se bendice con la concepción de nuestra hija que está por nacer.

A mis cuatro abuelos que desde donde están sé que se enorgullecen de mí y por la fuerza que me dieron en todo momento para terminar, cuando me preguntaban desesperadamente cuando culminaría la escuela, pues no deben de entender, por qué estos 5 años se hicieron tan largos.

A mi Hermano Duviel y mi cuñada Mildrey por recibirme en su casa con el amor y la alegría que los caracteriza, que me hacían olvidar toda tensión que me pudieran dar los estudios y aprovechar al máximo los días de vacaciones en familia.

A mi hermana Yuliet y mi hermano Exiquito, por estar pendientes todo el tiempo de mí y no perder un detalle de mi trayectoria.

A mis sobrinos Duvielito y Diego por ese amor que solo los niños saben dar.

A mi cuñado Becker por hacer feliz a la persona que más quiero en este mundo, por querer y ayudar a mi familia, en especial mis abuelos que es la alegría y tranquilidad más grande que pueda tener.

A mis tíos Mari, Chucha, Ortencia, Mayra y Bárbara e Ignacio por tanto cariño y preocupación.

A mis primos en general, que no menciono por lo extensa de la lista.

A mis suegros Ruwira y Bárbara, por acogerme en su familia como una hija más. A mi suegra por tantos cuidados y ayuda, por esa preocupación constante por mi estado para que todo terminara bien y a mi suegro por lo que ha hecho por mi estos años que no es poco y por poner esa cara cada vez que algo me iba mal que obligaba esforzarme más para no decepcionarlo.

A Tere por todas las atenciones que me ha dado, por protegerme como una hija, y por acogerme en su casa como miembro más de su familia independientemente de mi relación con Randy.

A mis cuñados Nanaris y Raulito y mi sobrina Patty, por hacer la casa más alegre y sacarme de la rutina de preocupaciones y estudios.

A los viejos Gonzalito y Tere, por ser conmigo de una manera especial que me hacía dichosa entre mucha gente.

A Guille por estar pendiente de mi carrera y ayudarme a solucionar a su manera, los imprevistos que eh tenido.

A mi compañera de Tesis Yelina que no tengo palabras como agradécele el esfuerzo que ha hecho para ayudarme a llevar a cabo esta difícil tarea.

A Noslen por toda la información brindada y estar presente para cada urgencia nuestra.

A Karel que con sus locuras hizo más agradables mis días aquí. Sin dudas me llevo un buen amigo.

A Daymelis por ser mi conciencia y cargar conmigo en todos los trabajos a pesar todo.

A los vecinos y esa inmensa familia de amigos que me ha dado esta universidad, muchos de ellos aquí presente.

Al tribunal por sus sugerencias y recomendaciones.

A mis tutores Yuniel y Yoan por su dedicación.

A todos ustedes. Gracias.

Yeline:

En primer lugar quiero agradecer a Dios porque sin él nada de esto hubiera sido posible, por haberme dado la sabiduría, el entendimiento y la fortaleza para poder llegar al final de mi carrera.

Mami, no me equivoco si digo que eres la mejor mamá del mundo, gracias por todo tu esfuerzo, tu apoyo y por la confianza que depositaste en mí. Gracias por tu paciencia y esas palabras sabias que siempre tienes para mis enojos, mis tristezas y mis momentos felices, por ser mi amiga y ayudarme a cumplir mis sueños. Gracias por estar siempre a mi lado. Te quiero mucho.

Palito no tengo palabras para agradecerte todo lo que has hecho por mí. Sin tus esfuerzos no hubiera podido llegar hasta aquí. Todos los días de mi vida le agradezco a Dios por haberme permitido tener un padre como tú, eres el mejor, te quiero mucho.

A mis hermanitos por ser mi fuente de inspiración, ustedes me motivan a ser mejor cada día y puedan estar orgullosos de su hermana mayor. Gracias por traer alegría a mi vida. Los quiero mucho.

A toda mi familia en general por el apoyo que siempre me han brindado.

A mis tíos Iraida y Andrés porque han estado conmigo en todos estos años, desde el inicio de mi carrera hasta el final, gracias por recibirme en su casa y tratarme como una hija más.

A mis primos Daylin y Andresito, los quiero como si fueran mis hermanos, con los que compartí muchos fines de semana e hicieron que estos años fueran mejores.

A mi dúo de tesis Dunia, gracias por ser mi amiga, por siempre estar a mi lado en los buenos y malos momentos, gracias por ayudarme hacer posible este sueño.

A Raciél aunque hayamos terminado nuestra relación, quiero agradecerte por todos los años que pasamos juntos. Estos años lejos de mi familia fueron difíciles, pero tú supiste llenar esos días de alegría y hacer que ese tiempo pasara más rápido.

A Dania por siempre estar pendiente de nosotras y ayudarnos en todos los preparativos de la tesis.

A todos mis amigos, quisiera nombrarlos a todos pero son muchos. Gracias por estar conmigo en todo este tiempo donde he vivido muchos momentos felices y tristes de mi vida. Por siempre estar ahí cuando más los he necesitado. Con cada uno de ustedes viví momentos especiales que nunca voy a olvidar, gracias por ser mis amigos y recuerden que siempre los llevaré en mi corazón, los quiero mucho.

A los tutores Proenza y Yoan que hicieron posible la realización de esta tesis, gracias por todo el tiempo que nos dedicaron y la paciencia que tuvieron para explicarnos una y mil veces las mismas cosas.

A mi grupo por aguantarme todos los días en el aula, no fue nada fácil.

A las niñas de mi apartamento con las que conviví en los últimos dos años, gracias por brindarme su ayuda en todo momento.

A todos los profesores que tuve en estos años, a quienes les debo gran parte de mis conocimientos.

Quiero agradecer a todas las personas en general, por el apoyo que siempre me han brindado en el transcurso de cada año de mi carrera universitaria.

Resumen

La agricultura es el arte de cultivar la tierra; son los diferentes trabajos de tratamiento del suelo y cultivo de vegetales, normalmente con fines alimenticios. El sector agrícola en Cuba, desempeña un importante papel en la economía nacional. Actualmente, el Ministerio de la Agricultura (MINAG), es la institución encargada de controlar este sector en el país. El mismo, presenta algunos problemas relacionados con la planificación y gestión de los recursos, los cuales se hacen a partir de la experiencia de los propios agricultores y de información almacenada en lugares específicos. Por tanto, muchos de los procesos que se realizan en esta área como: el riego, fumigación, control de plagas, rotación de cultivos, entre otros procesos esenciales, no son consultables ni públicos para ser analizados. Por la ausencia de sistemas informáticos que gestionen tales actividades, el control de estos procesos se realiza manualmente.

El presente trabajo tiene como objetivo fundamental desarrollar un Sistema de Información Geográfica (SIG) para la Gestión de Cultivos, que ayude a mejorar los problemas presentados en el sistema agrícola cubano. Para el desarrollo del software se decide usar las mismas herramientas y tecnologías definidas por la plataforma GeneSIG, para lograr una fácil integración con la misma.

Palabras claves: cultivo, sector agrícola, sistema de información geográfica.

Abstract

Agriculture is the art of cultivating the land; they include the different soil treatments and cultivation of vegetables, usually for food. The agricultural sector in Cuba, plays an important role in the national economy. Currently, the Ministry of Agriculture (MINAG), is the institution responsible for controlling this sector in the country. MINAG actually presents some problems related to the planning and management of resources, which are gathered from the experience of the farmers themselves and information stored in specific locations. Therefore, much of the information for the processes that take place in this area such as irrigation , spraying, pest control, crop rotation, among other essential processes are not accessible or not available to the public for analysis. In the absence of computerized information that manage such activities, control of these processes are done manually.

This work main objective is to develop a Geographic Information System (GIS) for crop management, to help improve the problems presented in the Cuban agricultural system. To develop the software we decided to use the same tools and technologies defined by the GeneSIG platform for easy integration.

Keywords: farming, agriculture, geography, information systems.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Agricultura	5
1.2.2 Cultivo	6
1.2.3 Parcela	7
1.2.4 Sistemas de Información Geográfica (SIG)	7
1.3 Estudio de las soluciones existentes	11
1.3.1 SIGPAC (Tejerina, 2014)	11
1.3.2 SIGA (Tejerina, 2014)	13
1.3.3 SIG para las decisiones en la producción cañera de Santiago de Cuba (Rojas Martínez, et al., 2014)	14
1.4 Análisis de las soluciones existentes	16
1.5 Herramientas y tecnologías para el desarrollo del sistema	17
1.5.1 Plataforma de Desarrollo GeneSIG 2	17
1.5.2 Metodología de Desarrollo de Software	18
1.5.3 Lenguaje de Modelado	22
1.5.4 Herramientas CASE	23
1.5.5 Servidor Web	24
1.5.6 Lenguajes de programación	24
1.5.7 Gestores de bases de datos	27
1.5.8 Entornos de desarrollo integrados (IDE)	27
1.5.9 Servidor de mapas. MapServer 6.0	28
1.6 Conclusiones del capítulo	29
Capítulo 2: Análisis y diseño del SIG para la Gestión de Cultivos	30
2.1 Introducción	30
2.2 Modelo del dominio	30
2.3 Especificación de los requisitos del sistema	31
2.3.1 Requisitos funcionales (RF) del sistema	32
2.3.2 Requisitos no funcionales (RNF) del sistema	36
2.4 Arquitectura del sistema	38

2.4.1 Patrones arquitectónicos.....	38
2.4.1 Estilo Arquitectónico.....	39
2.5 Patrones de diseño.....	39
2.5.1 Patrones GRASP	39
2.5.2 Patrones GOF.....	40
2.6 Modelo de diseño	41
2.6.1 Diagrama de clases del diseño	41
2.6.2 Descripción de las clases del diseño del RF Insertar cultivo	43
2.7 Diseño de la Base de Datos del SIG para la Gestión de Cultivos	44
2.7.1 Diagrama de clases persistentes	44
2.7.2 Diagrama Entidad – Relación.....	45
2.7 Propuesta de solución	46
2.8 Conclusiones del capítulo.....	46
Capítulo 3: Implementación del sistema y pruebas	47
3.1 Introducción.....	47
3.2 Diagrama de componentes.....	47
3.3 Diagrama de despliegue.....	48
3.4 Pruebas del sistema	48
3.4.1 Diseño de Casos de Prueba	49
3.5 Conclusiones del capítulo.....	54
Conclusiones	55
Recomendaciones	56
Referencia Bibliográfica	57
Anexos.....	61

Índice de Figuras

Fig 1: Representación visual del modelo ráster y vectorial en los SIG (Brenes, 2009).....	10
Fig 2: Pantalla del Visor con las Zonas de Trabajo visibles del SIGPAC	12
Fig 3: Pantalla del Visor con las Zonas de Trabajo visibles del SIGA.....	14
Fig 4: Mapa temático de las unidades empresariales de base en la provincia de Santiago de Cuba	15
Fig 5: Mapa temático de rango de rendimiento de la UEB Julio A. Mella.....	15
Fig 6: Modelo de dominio	30
Fig 7: Diagrama de Clases del Diseño Insertar Cultivo	42
Fig 8: Diagrama de Clases del Diseño Insertar parcela	42
Fig 9: Diagrama de clases persistentes	45
Fig 10: Modelo físico de la base de datos	45
Fig 11: Diagrama de componentes.....	47
Fig 12: Diagrama de despliegue	48
Fig 13: Prueba de caja negra.....	53
Fig 14: Interfaz principal del SIG para la Gestión de Cultivos	62
Fig 15: Interfaz del SIG para Gestionar parcela	63
Fig 16: Interfaz del SIG para Gestionar cultivo	63
Fig 17: Fig 17: Interfaz del SIG para sembrar un cultivo en una parcela determinada	64
Fig 18: Interfaz del SIG para mostrar los cultivos afectados por una plaga determinada.....	64

Índice de Tablas

Tabla 1: Descripción de las fases (Sánchez, 2015).....	19
Tabla 2: Descripción de las Disciplinas Variación AUP – UCI (Sánchez, 2015).....	20
Tabla 3: Descripción de las clases del modelo del dominio	31
Tabla 4: Descripción de las clases del diseño	43
Tabla 5: Caso de prueba Insertar Cultivo	50
Tabla 6: Descripción de las variables del Caso de Prueba Insertar Cultivo.....	52

Introducción

El surgimiento de la agricultura fue un paso clave en el desarrollo de la humanidad y se convirtió en una actividad económica fundamental para la subsistencia de la población. Se basa en la labranza y cultivo de la tierra o domesticación de plantas y animales, e incluye todos los trabajos relacionados con el tratamiento del suelo y la plantación de cultivos. La misma ha evolucionado significativamente transitando desde los cultivos en las tierras fértiles del oeste de Asia, en las márgenes del Nilo, hasta la tecnología empleada en la actualidad.

Desde el pasado siglo, la agricultura se caracterizó por un aumento de productividad, disminución de la fuerza de trabajo, aplicación de fertilizantes artificiales o sintéticos, aplicación de pesticidas, cría selectiva y mecanización. Tal situación se ha reflejado en nuestra economía que ha transitado de una agricultura de supervivencia, al monocultivo de la caña de azúcar y en estos momentos se trata de diversificar las producciones agropecuarias para poder satisfacer las demandas de la población.

Las carencias de productos e incumplimientos en la adquisición de materias primas, ha afectado considerablemente el cumplimiento de los planes productivos. No obstante, se hace uso de las Tecnologías de la Información y la Comunicación (TICs) para apoyar el desarrollo agrícola. Para ello, cada vez se destinan más recursos en la creación de sistemas de información que permitan obtener un mejor análisis de sus datos, mediante la aplicación de los SIG.

Actualmente, el proceso de planificación se realiza a partir de la experiencia de los propios agricultores en el caso de los campesinos independientes. Dicha experiencia, se incluye en la carta tecnológica o Instructivo Técnico del Cultivo como hoy día se le denomina. Este documento no está informatizado, o sea, está localmente planificado por cada entidad y es utilizado en las cooperativas y demás instituciones agrícolas de Cuba, con el fin de regular la planificación de los recursos necesarios para asegurar el plan de producción. Establece qué mecanismos utilizar para que esos recursos, después de producidos, lleguen a su destino y puedan ser una verdadera garantía para la ejecución y el cumplimiento del plan.

La carencia de sistemas informáticos para el proceso agrícola y gestión de cultivos, obliga a registrar toda la información en archivos de papel y formato duro. Por tanto, muchos de los períodos establecidos para el riego, fumigación, control de plagas, así como la rotación de cultivos y otros procesos esenciales, a pesar de que están reglamentados, no son consultables ni públicos para su análisis. El control de estos procesos

se realiza de forma manual. Por otro lado, las personas encargadas de realizar éstas tareas, no cuentan con una planificación detallada para cada tipo de cultivo, ni formas efectivas para el aviso en caso de retrasos; elementos esenciales para mejorar la calidad del producto a entregar.

La realización de actividades por separado, incluso por personas distintas que no pueden comunicar sus resultados o actividades al resto de los encargados, no permite la transmisión de conocimientos. Sumado a esto, no se posee la geo-referenciación exacta de los campos y/o parcelas, ni los tipos de cultivos por área.

El empleo de sistemas informáticos, específicos o integradores, desarrollados a demanda del interesado para estos fines, ayudaría a mejorar los tiempos de respuesta ante alguna afectación, la toma de decisiones ante situaciones excepcionales, e incluso mejorar el proceso de asignación de recursos teniendo en cuenta distancias, características zonales, entre otros aspectos de interés.

Debido a lo planteado anteriormente se identifica el **problema a resolver**: ¿Cómo mejorar la gestión de los cultivos en el sistema agrícola cubano?

Para darle solución al problema planteado se define como **objeto de estudio**: El Proceso Agrícola en Cuba.

Delimitando como **campo de acción**: La Gestión de Cultivos en cooperativas e instituciones agrícolas cubanas.

Se plantea como **objetivo general**: Desarrollar un sistema para la representación y el análisis geoespaciales de información sobre los cultivos, que pueda desplegarse en las instituciones agrícolas cubanas.

Para darle solución al objetivo trazado se definen las siguientes **tareas de investigación**:

1. Realizar un análisis de las causas que originan la mala planificación y gestión de los cultivos en Cuba.
2. Analizar las características y limitaciones de las soluciones existentes referente a los cultivos tanto en el ámbito internacional como nacional.
3. Caracterizar las herramientas y tecnologías seleccionadas para la implementación de este tipo de sistemas.
4. Realizar los artefactos y la documentación según la metodología de desarrollo seleccionada.
5. Implementar el SIG para la Gestión de Cultivos en Cuba.

6. Evaluar el sistema implementado mediante pruebas para garantizar su correcto funcionamiento.

Para guiar el proceso de investigación se definen las siguientes **preguntas científicas**:

1. ¿Cuáles son los problemas presentados en las cooperativas e instituciones cubanas en el proceso agrícola?
2. ¿Cómo influiría la mejora del proceso de gestión agrícola en la disminución de las pérdidas de cultivos o productos, los tiempos de ejecución de las tareas y la disminución de errores en la asignación de recursos?
3. ¿Cuáles son las características que debe cumplir el sistema para mejorar los problemas existentes en las cooperativas e instituciones cubanas, asociados a la gestión agrícola?
4. ¿Con la implementación de un SIG se logrará mejorar la planificación y gestión de los recursos en el sistema agrícola cubano?

Para desarrollar la investigación se utilizaron los siguientes **Métodos de Investigación Científica**, los mismos pueden clasificarse en métodos teóricos y empíricos, los cuales están dialécticamente relacionados entre sí.

Métodos Teóricos:

1. Analítico-sintético

“El análisis de un objeto se realiza a partir de la relación que existe entre los elementos que conforman dicho objeto como un todo; y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis” (Hernández León, y otros, 2011). Este método ayudó a sintetizar el análisis realizado a toda la documentación consultada para el desarrollo de la investigación relacionada con la agricultura, haciendo énfasis en la gestión de cultivos. Dicho análisis parte de la relación entre los elementos que conforman el proceso agrícola y la síntesis se produce basándose en los resultados previos a ese análisis.

2. Análisis histórico-lógico

“Está vinculado al conocimiento de las distintas etapas de los objetos en su sucesión cronológica; para conocer la evolución y desarrollo del objeto o fenómeno de investigación se hace necesario revelar su historia. Los métodos lógicos se basan en el estudio histórico, poniendo de manifiesto la lógica interna de desarrollo, de su teoría y halla el conocimiento más profundo de esta, de su esencia” (Hernández León, y otros, 2011). Este método se utilizó para realizar un estudio desde el año (1990 - 2015) tanto a nivel nacional como internacional de los diferentes SIG destinados a la agricultura. Se estudiaron algunos de los

implementados hasta la actualidad, así como la evolución de los mismos, sus principales características, ventajas y desventajas, sirviendo de referencia para la investigación e implementación del sistema a desarrollar.

Métodos empíricos

1. Entrevista

“La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación “ (Hernández León, y otros, 2011). Este método se utilizó para identificar los requerimientos y funcionalidades que debe cumplir el sistema a implementar, en base a las necesidades y expectativas del cliente. Estas entrevistas se le realizaron a varios especialistas del MINAG y trabajadores de la agricultura. Ver anexo 1.

La estructura de este trabajo de investigación, luego de la introducción es la siguiente:

Capítulo 1: Fundamentación teórica.

En este capítulo se exponen los principales conceptos relacionados con la investigación. Se describen términos técnicos de importancia y se realiza un estudio a sistemas con características similares al sistema que se desea implementar. Se define la metodología para el desarrollo del software, así como las principales herramientas y tecnologías que se van a usar en el desarrollo del sistema.

Capítulo 2: Análisis y diseño del SIG para la Gestión de Cultivos.

En este capítulo se desarrolló la solución propuesta teniendo en cuenta la metodología seleccionada. Se definen los requisitos funcionales y no funcionales con los que debe cumplir el sistema para su correcto funcionamiento, así como la arquitectura y el patrón arquitectónico que se va a usar para la implementación del mismo. Además se obtienen todos los artefactos generados en el diseño.

Capítulo 3: Implementación del sistema y pruebas.

En este capítulo se obtiene el diagrama de componentes y despliegue correspondiente al sistema. Además, se describen los procesos de implementación y pruebas realizadas al software.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En este capítulo se hace un análisis del objeto de estudio para comprender el funcionamiento del Proceso Agrícola en Cuba. Se realiza un estudio de sistemas similares al que se propone implementar y se presenta un resumen de varias soluciones que brindan de alguna manera respuesta al problema científico planteado, caracterizando las ventajas e insuficiencias de las mismas, así como sus principales funcionalidades. Por último se definen las herramientas y tecnologías que se utilizarán en el desarrollo de la aplicación.

1.2 Conceptos asociados al dominio del problema

Para una mejor comprensión de lo planteado en la situación problemática es importante describir algunos conceptos que se utilizan en la investigación:

1.2.1 Agricultura

“La agricultura se trata de toda una serie de relaciones nuevas entre el hombre, la tierra, la vegetación y los animales. Implica la transición a un ecosistema totalmente diferente y comporta un nuevo orden de relaciones estructurales entre los participantes. Con la agricultura se crearon las bases económicas y las situaciones sociales propicias para el surgimiento de las sociedades estatales. La capacidad de producir alimentos, les permitió aumentar su control sobre la naturaleza y multiplicarse” (Ravelo Ortega, y otros, 2007).

La agricultura es la actividad agraria que comprende todo un conjunto de acciones humanas que transforma el medio ambiente natural, con el fin de hacerlo más apto para el crecimiento de las siembras. Es el arte de cultivar la tierra, refiriéndose a los diferentes trabajos de tratamiento del suelo y cultivo de vegetales, normalmente con fines alimenticios (Ravelo Ortega, y otros, 2007).

Según (Ravelo Ortega, y otros, 2007) los tipos de alternativas en la agricultura son las siguientes:

1. **Agricultura tradicional:** Sistema de producción basado en conocimientos y prácticas indígenas, que han sido desarrollados a través de muchas generaciones.
2. **Agricultura convencional:** Sistema de producción agropecuaria en la que se utilizan sustancias químicas sintéticas de manera parcial o total.
3. **Agricultura de secano:** Esta forma de agricultura, depende totalmente de las lluvias y no se emplea el riego en ningún momento. La fecha de siembra del cultivo estaría sujeta al comienzo de las lluvias.

4. **Agricultura intensiva:** Forma de aprovechamiento del suelo. Se caracteriza por los altos rendimientos debido a la utilización de sistemas mecánicos, regadío, rotación de cultivos y técnicas científicas (aunque puede haber agricultura intensiva con técnicas tradicionales).
5. **Agricultura extensiva:** Forma de aprovechamiento del suelo. Se caracteriza por sus bajos rendimientos. Suele ser una agricultura de secano y en suelos pobres.
6. **Agricultura orgánica:** Esta clase de agricultura es muy antigua y una de las más sanas para el suelo y su entorno.
7. **Agricultura sostenible:** El objetivo de la agricultura sostenible es producir alimentos de manera eficaz y productiva, pero conservando y mejorando el medio ambiente y las comunidades locales.
8. **Agricultura ecológica:** La agricultura ecológica es un método de producción que vela por la conservación del medio ambiente y respeta al máximo los ciclos biológicos, sólo se utilizan productos naturales.
9. **Agricultura de precisión:** La agricultura de precisión permite a través de determinadas herramientas dar a cada zona del campo cultivado el tratamiento agronómico más apropiado, tanto desde el punto de vista agronómico-productivo como del ambiental.
10. **Agricultura en casa de cultivo:** Esta forma de explotación de la agricultura, llamada también cultivo protegido, ha tomado fuerza en Cuba en los últimos 5 años y se ha extendido a escala nacional. Consiste en una casa con estructura metálica y dimensiones (ancho, largo y alto, en dependencia de sus valores será la superficie cultivada

1.2.2 Cultivo

“Los cultivos se suceden sobre una misma parcela y dejan el suelo en condiciones físicas, químicas y biológicas muy diferentes, haciendo aparecer así la importante noción de precedente cultural. Los mismos pueden influir sobre los que le sigan a través de la cantidad de agua o de nutrientes utilizables que dejan en el suelo por intermedio de sus residuos los cuales pueden ser beneficiosos o perjudiciales” (Ravelo Ortega, y otros, 2007).

Según (Ravelo Ortega, y otros, 2007) dentro de las técnicas aplicadas en las empresas agrícolas se encuentra la rotación de cultivos, la cual consiste en alternar plantas de diferentes familias y con necesidades nutritivas diferentes en un mismo lugar durante distintos ciclos. De esta manera, se evita que el suelo se agote y que las enfermedades que afectan a un tipo de plantas se perpetúen en un tiempo determinado. Además, se utiliza la alternativa, que no es más que distribución de las especies económicas en una unidad de producción, año por año y de manera tal que, por lo menos, de un año a otro no se repitan sucesivamente en las mismas áreas de suelo.

1.2.3 Parcela

En la agricultura se denominan parcela a una porción pequeña de terreno que se va a cultivar. La división del campo en parcelas ofrece muchas ventajas, sirve para controlar fácilmente el trabajo realizado por los operarios, ya sea por normas o sin normas (Ravelo Ortega, y otros, 2007).

Según (Ravelo Ortega, y otros, 2007) es posible analizar la parcela desde distintas perspectivas:

- La superficie bruta es la superficie original de acuerdo a los datos que consten en el catastro¹, el registro de propiedad y las mediciones reales.
- La superficie neta, en cambio, es la superficie que no se encuentra reservada para la ubicación de dotaciones urbanísticas públicas.
- La parcela mínima, es la superficie que tiene que tener una parcela para que las autoridades puedan autorizar los usos permitidos según el planeamiento urbano
- Las parcelas resultantes son aquellas formadas a partir del desarrollo de otras actuaciones.

1.2.4 Sistemas de Información Geográfica (SIG)

“Un conjunto de hardware, software y datos geográficos para capturar, manipular, analizar y mostrar información geográficamente referenciada” (ESRI, 1995).

“Los SIG se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente en función de capturar, almacenar, analizar, transformar, además presentar toda la información geográfica y de sus atributos con el fin de resolver problemas complejos de planificación y gestión” (Brenes, 2009).

¹ Catastro (registro estadístico para determinar la extensión geográfica y riqueza de alguna demarcación)

“Un SIG consiste en información de naturaleza diversa sobre un determinado territorio, almacenada en un conjunto de BD² tanto gráficas como alfanuméricas, cuya relación con el territorio se realiza a través de un sistema de referencia geográfico y se gestiona a través de uno o varios programas informáticos específicos; el conjunto es soportado por un sistema de computadores y por un personal especializado” (Fallas, 2011).

“Un SIG puede ser concebido como una especialización de un sistema de bases de datos, caracterizado por su capacidad de manejar datos geográficos, que están geo-referenciados y los cuales pueden ser visualizados como mapas” (Braken, 1992).

Debido a la gran diversidad de definiciones propuestas para los SIG y luego de estudiar varias de ellas, el autor de la presente investigación asume la propuesta por el NCGIA³: *“Un SIG es 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 de planificación y gestión” (NCGIA, 1990).*

Estos sistemas son una tecnología que permite gestionar y analizar la información espacial. Funcionan como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos, e inversamente preguntando por un registro de la base de datos se puede saber su localización en la cartografía (Brenes, 2009).

Los SIG permiten separar la información en diferentes capas temáticas y las almacena independientemente. Las mismas permiten trabajar con ellas de manera rápida y sencilla, facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma (Brenes, 2009).

Componentes que conforman un SIG:

Según (Olaya, 2010) la composición de los SIG está dividida en cinco elementos conceptuales: hardware, software, datos, personal y métodos:

² BD (Base de datos)

ESRI (Environmental Systems Research Institute)

³ NCGIA (National Center for Geographic Information and Analysis)

- **Hardware:** este elemento requiere varios tipos de dispositivos, desde equipos computadores centralizados para el almacenamiento de los datos, hasta capturadores de información geográfica. Su organización ha de determinarse por un hardware específico y de buenas prestaciones para cumplir con las necesidades de las aplicaciones asociadas al procesamiento y análisis de los datos que se representan en el SIG.
- **Software:** este incluye no solo los programas de SIG, sino también los programas informáticos de BD (Base de Datos), estadísticos, de procesamiento de imágenes y cualquier otro software.
- **Datos:** propician toda la información geográfica para la creación de un SIG.
- **Personal:** este es el componente más importante de un SIG. El personal debe desarrollar los procedimientos y definir las tareas para desarrollarlo.
- **Métodos:** el análisis requiere métodos bien definidos y consistentes para producir resultados correctos y reproducibles.

Según (Brenes, 2009) las principales características de los SIG son las siguientes:

- La capacidad de visualización de información geográfica compleja a través de mapas.
- La funcionalidad de los SIG como una BD sofisticada, en la que se mantiene y relaciona información espacial y temática.
- La diferencia con las BD convencionales radica en que toda la información contenida en un SIG está unida a entidades geográficamente localizadas. Por ello, en un SIG la posición de las entidades constituye el eje del almacenamiento, recuperación y análisis de los datos.
- Representan una tecnología de integración de información.
- Se han desarrollado a partir de innovaciones tecnológicas sobre campos especializados de la geografía y otras ciencias (tratamiento de imágenes, análisis fotogramétricos, cartografía automática, entre otras), para constituir un sistema único más potente que la suma de las partes.
- Permiten unificar la información en estructuras coherentes.
- Este carácter integrador y abierto, hace de los SIG un área de contacto entre variados tipos de aplicaciones informáticas, destinadas al manejo de información con propósitos y formas diversas; por ejemplo: programas estadísticos, gestores de bases de datos, programas gráficos, hojas de cálculo, procesadores de texto, entre otros.

- Los límites y diferencias entre los SIG se enfocan en, los programas de diseño asistido por computadoras, los de cartografía temática y los de tratamiento de imágenes que son especialmente difusos. Aunque sus diferencias estriban sobre todo en el modelo de datos y en las capacidades de análisis de información espacial.

Los SIG funcionan con dos tipos diferentes de información geográfica, Modelo vectorial y Modelo ráster (Olaya, 2010).

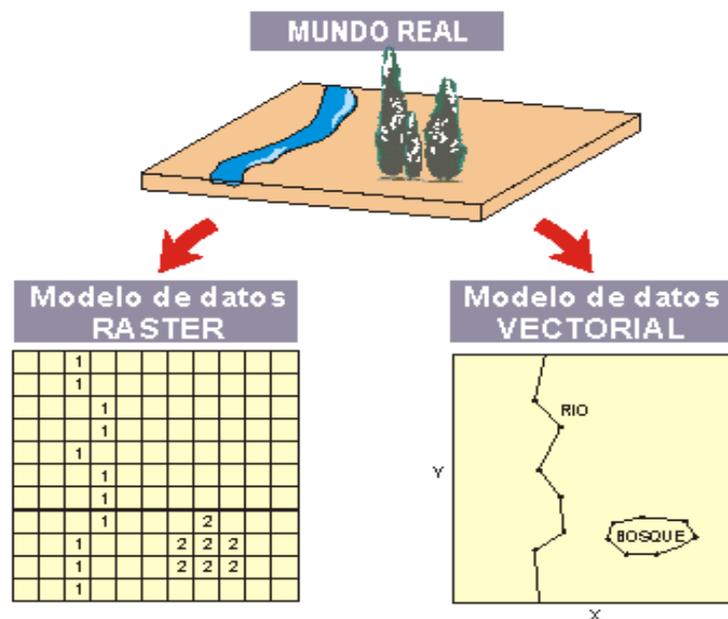


Fig 1: Representación visual del modelo ráster y vectorial en los SIG (Brenes, 2009)

Modelo vectorial: es una estructura de datos utilizada para almacenar datos geográficos. Los datos vectoriales constan de líneas o arcos, definidos por sus puntos de inicio y fin, y puntos donde se cruzan varios arcos, los nodos. La localización de los nodos y la estructura topológica se almacena de forma explícita. Las entidades quedan definidas por sus límites solamente y los segmentos curvos se representan como una serie de arcos conectados. El almacenamiento de los vectores implica el almacenamiento explícito de la topología, sin embargo solo almacena aquellos puntos que definen las entidades y todo el espacio fuera de éstas no está considerado (Olaya, 2010).

Modelo ráster: es un método para el almacenamiento, el procesado y la visualización de datos geográficos. Cada superficie a representar se divide en filas y columnas, formando una malla o rejilla regular. Cada celda ha de ser rectangular, aunque no necesariamente cuadrada. Cada celda de la rejilla guarda tanto las coordenadas de la localización como el valor temático. La localización de cada celda es implícita, dependiendo directamente del orden que ocupa en la rejilla, a diferencia de la estructura vectorial en la que

se almacena de forma explícita la topología. Las áreas que contienen idéntico atributo temático son reconocidas como tal, aunque las estructuras ráster no identifican los límites de esas áreas como polígonos en sí (Olaya, 2010).

1.3 Estudio de las soluciones existentes

Debido a la mala planificación y gestión de los recursos, se tienen que lamentar pérdidas de cultivos y productos agrícolas, ocasionando pérdidas a la economía del país. Los SIG se han convertido en una solución factible para resolver estos problemas complejos, ya que permiten realizar el análisis y procesamiento de datos referenciales, lo que contribuye al proceso de análisis para la toma de decisiones, ayudando a realizar una mejor planificación y gestión de los recursos.

Para darle solución a la problemática descrita en la introducción, se realizó un análisis de algunas soluciones existentes encontradas que sirvieran como base para la solución que se desea implementar. De estos sistemas se estudiaron las principales características, funcionalidades, ventajas y desventajas que presentan los mismos. A partir de la investigación realizada se lograron identificar los siguientes sistemas: Sistemas de Información Geográfica de Parcelas Agrícolas (SIGPAC), Sistemas de Información Geográfica de Datos Agrario (SIGA) y SIG para las decisiones en la producción cañera de Santiago de Cuba. A continuación se caracterizan los sistemas antes mencionados.

1.3.1 SIGPAC (Tejerina, 2014)

Es un sistema de identificación de parcelas agrarias, creado a instancia de la Comisión Europea. Permite identificar geográficamente las parcelas declaradas por los agricultores y ganaderos, en cualquier régimen de ayudas relacionado con la superficie cultivada o aprovechada por el ganado en todo el territorio español. Este sistema se ha convertido en una herramienta de enorme utilidad en campos diferentes del agrario (geología, infraestructuras, urbanismo, etc.), debido a su concepción y desarrollo, en el que se hace uso continuo y permanente de las tecnologías más avanzadas en información geográfica automatizada.

El sistema consta de un mosaico de ortofotos⁴ digitales que abarcan toda España, de forma que, para cada referencia concreta, el sistema proporciona automáticamente la imagen en pantalla de la parcela referenciada, permitiendo asimismo su impresión en papel.

⁴ Ortofotos: (presentación fotográfica de una zona de la superficie terrestre, en la que todos los elementos presentan la misma escala, libre de errores y deformaciones, con la misma validez de un plano cartográfico).

El SIGPAC es un software privativo publicado bajo la licencia Creative Common - Reconocimiento (CC-by 3.0). La versión completa del Visor SIGPAC está desarrollada sobre tecnología SilverLight de Microsoft, siendo necesaria la versión 3 de Silverlight para el correcto funcionamiento del Visor.

Entre los objetivos más importantes del Proyecto SIGPAC figuran los siguientes:

- Facilitar a los agricultores la presentación de solicitudes, mediante la producción de los soportes gráficos necesarios para las declaraciones de superficie.
- Facilitar los controles administrativos ya que, la información digital ayudará a la administración a identificar mejor el origen de los errores derivados de las declaraciones de los agricultores o de la grabación de los datos, y servirá de soporte documental para la resolución de casos dudosos detectados como resultado de estos controles.
- Facilitar los controles sobre el terreno, agilizando la localización de parcelas y permitiendo la realización de "visitas rápidas " tanto en los controles clásicos como de teledetección.

En la Fig 2 se presenta la interfaz principal del SIGPAC

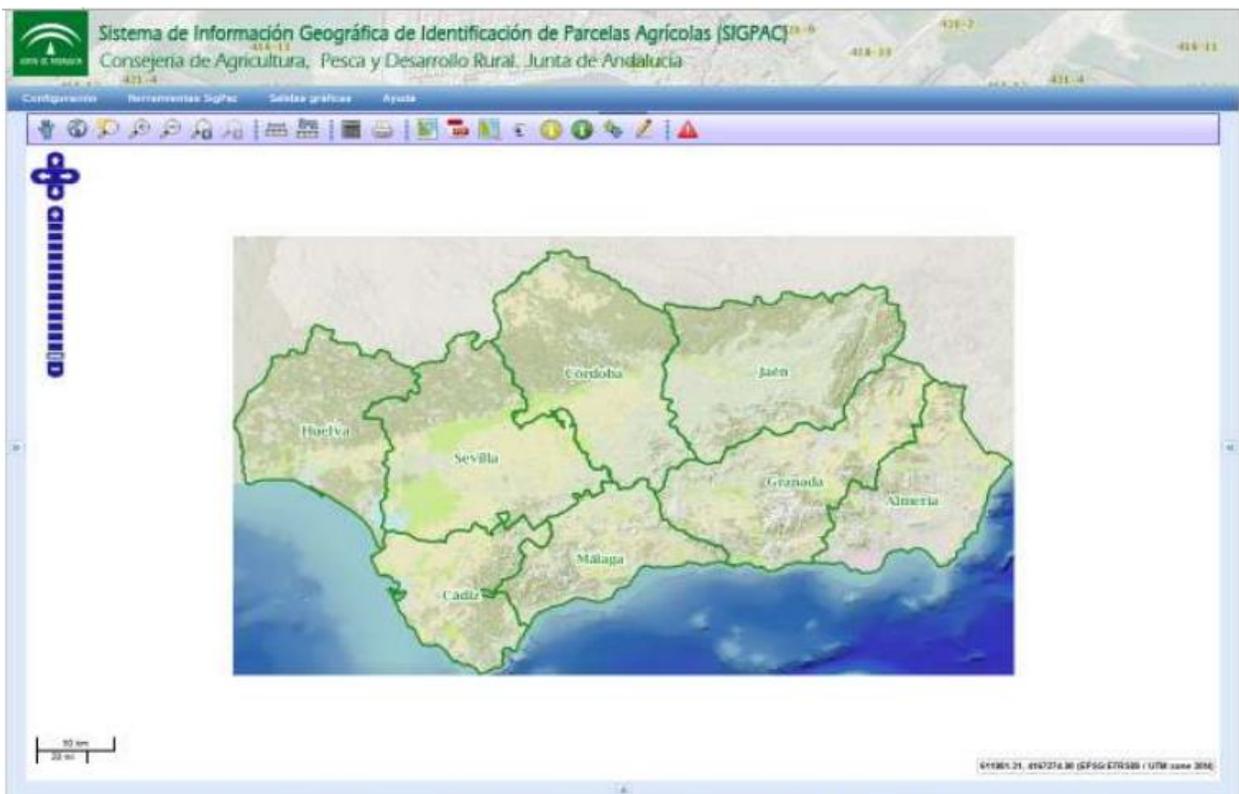


Fig 2: Pantalla del Visor con las Zonas de Trabajo visibles del SIGPAC

1.3.2 SIGA (Tejerina, 2014)

Es un sistema mediante el cual se puede navegar, analizar y consultar la información geográfica relativa de datos de tipo agrario. Esta aplicación está desarrollada sobre una arquitectura totalmente orientada a servicios WEB. Es un software privativo, publicado bajo la Licencia Creative Common -Reconocimiento (CC-by 3.0). Permite ofrecer información cartográfica y alfanumérica englobada en dos aplicaciones:

Aplicación SIGCH⁵:

- Cartografía general: Salvo que se indique lo contrario, la cartografía es de elaboración propia.
- Mapas temáticos sobre variables agroclimáticas: Una de las capas de información existentes muestra las estaciones meteorológicas empleadas.
- Plan de Regionalización Productiva de España.
- Informes sobre Municipios: Se puede obtener datos sobre los municipios.

Aplicación MCA⁶

- Mapa de Cultivos y Aprovechamientos de España, a escala 1/50.000 m: ofrece información cartográfica y alfanumérica. Las consultas podrán realizarse para todo el territorio nacional por hojas y por municipios.
- Metodología utilizada: Impulsada por la Subdirección General de Cultivos Herbáceos, y con vistas tanto a poner al día el contenido del MCA como a hacerlo compatible con otras cartografías

En la Fig 3 se muestra la interfaz principal del SIGA

⁵ SIGCH (SIG relacionado con la O.C.M. de Cultivos Herbáceos)

⁶ MCA (Mapa de Cultivos y Aprovechamientos de España)

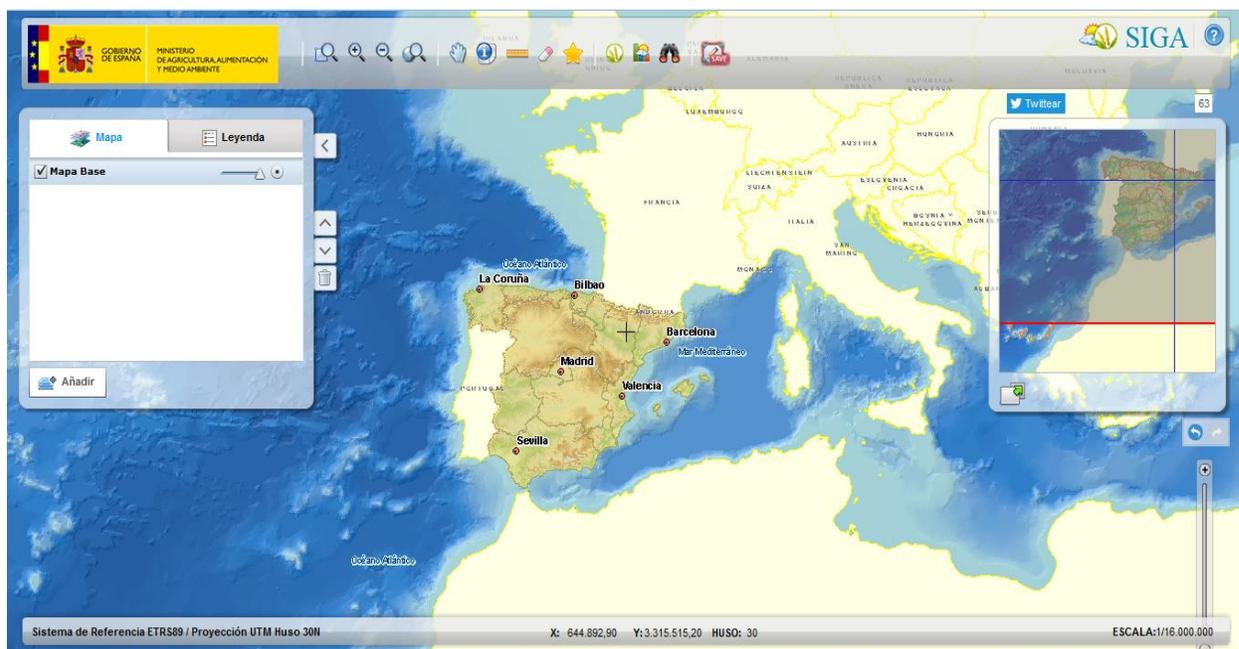


Fig 3: Pantalla del Visor con las Zonas de Trabajo visibles del SIGA

1.3.3 SIG para las decisiones en la producción cañera de Santiago de Cuba (Rojas Martínez, et al., 2014)

En la provincia de Santiago de Cuba se desarrolló un sistema geoespacial a través de informaciones temáticas sobre catastro, suelo, cultivos, geomorfología, relieve, hidrología, vegetación e infraestructura sobre MAPINFO, con el objetivo de mejorar la planificación agrícola y al acceso a información actualizada, de forma ágil y en tiempo real, para la toma de decisiones. Se utilizaron como instrumentos básicos la cartografía analógica existente, bases de datos de los servicios implementados en AZCUBA y otras de información complementaria. La zona de estudio comprende aproximadamente una superficie total de 83296.1 ha y 41557.29 ha dedicadas a la caña de azúcar, pertenecientes a las cinco unidades empresariales de base del Grupo Azucarero (AZCUBA) en la provincia SC; 56 unidades productoras y cinco bancos de semillas registradas.

Como resultado de la implementación del SIG en la provincia Santiago de Cuba a partir de la vinculación de la base cartográfica digitalizada con las bases de datos agrícolas. Se generó un grupo de mapas temáticos y gráficos con sus tablas asociadas; entre estos se encuentran: comportamiento varietal, prueba de validación comercial de variedades, prueba de fuego de variedades, estructura provincial por empresa y unidades, uso actual de los suelos, fertilización de los suelos, situación fitosanitaria de la provincia, la empresa y unidad, análisis del comportamiento de la maleza, de la tecnología de riego, lotes cañeros, comportamiento de los rendimientos cañeros, cobertura de los centros de limpieza, composición por cepa,

programación de cosecha, frente de corte, entre otros. Estos mapas constituyen el soporte para el desarrollo de una agricultura de precisión y permitieron tomar decisiones acertadas, de forma ágil y precisa.

Algunos ejemplos de mapas temáticos que ofrece esta aplicación son los siguientes:

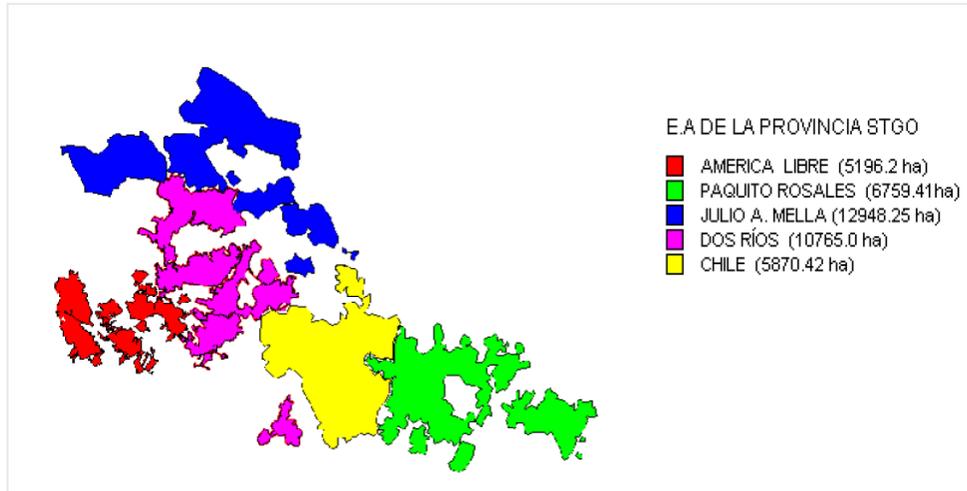


Fig 4: Mapa temático de las unidades empresariales de base en la provincia de Santiago de Cuba

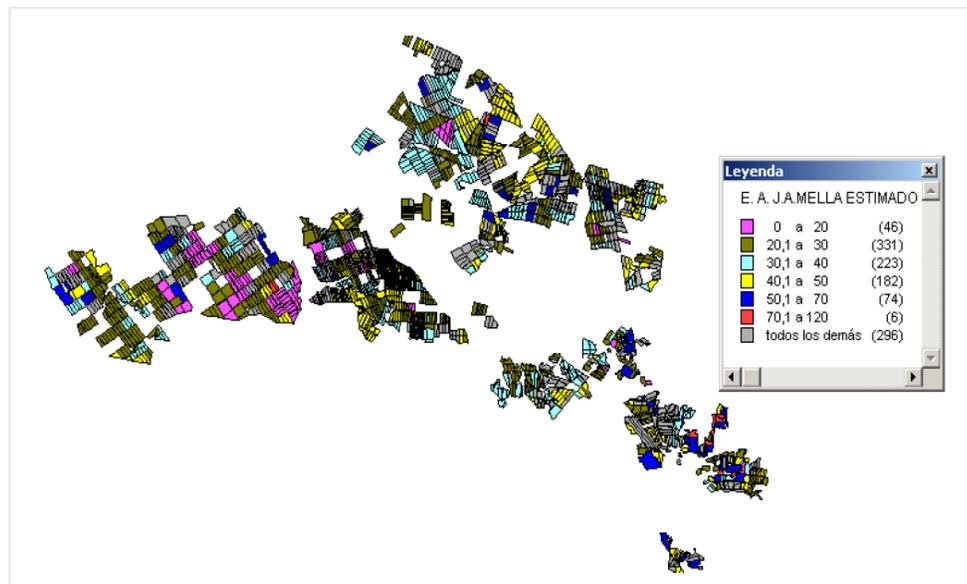


Fig 5: Mapa temático de rango de rendimiento de la UEB Julio A. Mella

Ventajas que ofrece la implementación del SIG para las decisiones en la producción cañera de Santiago de Cuba:

- Se cuenta con un grupo de ordenamiento territorial constituido por especialistas de la ETICA y la empresa azucarera; este grupo posibilita el seguimiento y la actualización del SIG.
- Las decisiones se toman de forma ágil y segura, lo cual permite incrementar la eficiencia y eficacia del proceso productivo, además de humanizar el trabajo del hombre.
- Elevar el nivel de técnicos, especialistas e investigadores en la tecnología desarrollada a través de cursos, seminarios, talleres y entrenamientos.
- Determinar y corregir de forma precisa los factores limitantes del suelo.
- Trazado de estrategias que permiten la disminución de pérdidas provocadas por plagas y enfermedades.
- Tratamiento al uso adecuado de los suelos con la aplicación de tecnologías que se adapten a las condiciones de cada suelo, relieve y clima.

1.4 Análisis de las soluciones existentes

Tanto el SIGA como el SIGPAC son software que fueron realizados con herramientas privativas y el SIG para las decisiones en la producción cañera solo crea mapas temáticos sobre algunos aspectos de interés en la provincia de Santiago de Cuba, relacionado con la producción cañera, por tanto, no son la base para el producto final.

A pesar, de no resolver en su totalidad los problemas actuales del sistema agrícola cubano, estos sistemas son una referencia para la realización de este trabajo, ya que ofrecen una guía sobre las funciones que pueden ser desarrolladas en el mismo aportando a la solución, la forma de representar las parcelas sobre el mapa de forma poligonal. El SIGPAC sirvió como guía para la gestión de parcelas, así como el SIGA para la gestión de cultivos, brindando toda la información necesaria respecto al desarrollo de estos procesos, teniendo en cuenta la localización de parcelas y cultivos por coordenada y la opción de poder generar un informe de los mismos. El SIG para las decisiones en la producción cañera ayudó a un mejor entendimiento del uso y funcionamiento de mapas temáticos.

A partir del análisis realizado a las soluciones existentes, se concluyó que no existe un sistema que integre todas las funcionalidades, que resuelvan los problemas descritos en la introducción. Por lo que, se decide implementar un SIG para la Gestión de Cultivos que ayude a mejorar los problemas existentes en el sistema agrícola cubano. En el siguiente epígrafe se describen las herramientas y tecnologías que se utilizarán para la implementación de la aplicación.

1.5 Herramientas y tecnologías para el desarrollo del sistema

Para el desarrollo del sistema se usaron las herramientas y tecnologías definidas por la plataforma GeneSIG, para lograr una fácil integración con la misma.

1.5.1 Plataforma de Desarrollo GeneSIG 2

La plataforma GeneSIG es una herramienta que se creó con el objetivo de facilitar y agilizar la creación de los SIG. Permite realizar representaciones y análisis de información referenciada geográficamente. Posee una estructura basada en plugins, lo que la convierte en una plataforma con un alto grado de interoperabilidad, debido a que permite agregar o quitar componentes de manera sencilla. De igual modo, permite separar los servidores de Base de Datos y Web en dos estaciones de trabajo diferentes, balanceando la carga del sistema, aumentando su disponibilidad y disminuyendo la posibilidad de fallas (Espinosa, y otros, 2012).

GeneSIG incluye un módulo encargado del tratamiento de la información, teniendo en cuenta criterios de roles y permisos sobre los recursos que maneja. Es un producto totalmente desarrollado y logrado sobre software libre, que permite la personalización de sus funcionalidades a cualquier negocio que lo requiera a través de la reutilización de sus componentes. Brinda servicios de georeferenciación y localización de recursos, así como la inclusión de datos y ubicación de nuevos objetos sobre mapas (Espinosa, y otros, 2012).

Cumple y respeta la estructura y el diseño de CartoWeb, pero solo utiliza los paquetes a los cuales le realizara cambios o aportes funcionales. Presenta también un abanico bastante completo de características propias de un geoportal, con posibilidad de ir añadiendo o desarrollando nuevos plugins. A través de los mismos, GeneSIG posee un amplio conjunto de funcionalidades, que actúan como herramientas de la misma plataforma y le brindan la posibilidad de ser altamente modular y escalable. El sistema se basa en la arquitectura cliente-servidor sobre plataforma Web, donde cada instancia del sistema en el cliente es independiente de la ejecución de otra (Espinosa, y otros, 2012).

Entre sus principales módulos se encuentra:

- Módulo de navegación.
- Módulo de selección.
- Módulo de Consulta Espacial.
- Módulo de Análisis.
- Módulo de Configuración del Mapa.
- Módulo de Impresión.

- Módulo de Catalogo.
- Módulo de Servicios.
- Módulo de Edición.
- Módulo de Ayuda.

1.5.2 Metodología de Desarrollo de Software

Una metodología de software se puede explicar como un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. La selección de una metodología de desarrollo varía dependiendo de los objetivos, tiempo y recursos con los que cuenta el equipo de desarrollo. Analizando las características principales de las metodologías utilizadas en la industria, se puede hacer referencia a ellas en dos grupos diferentes: Metodologías tradicionales y Metodologías Agiles (Sommerville, 2005).

Dentro de las Metodologías tradicionales se encuentra la metodología RUP (Proceso Unificado de Software). Para guiar el proceso de desarrollo de software se hace uso de la metodología AUP (Agile Unified Process), versión simplificada de RUP, por las características que presenta la misma y por ser la utilizada en el proyecto, permitiendo una mejor compatibilidad en la documentación. La misma describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (Sánchez, 2015).

Variación de AUP para la UCI

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la universidad.

➤ Descripción de las fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos en la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, titulada Ejecución y se agrega la fase de Cierre. Para una mejor comprensión se muestra la siguiente tabla 1 (Sánchez, 2015).

Tabla 1: Descripción de las fases (Sánchez, 2015)

Fases de AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	En esta fase se realiza un estudio inicial de la organización cliente que permite obtener una información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
Construcción		
Transición		
	Cierre	En esta fase se analizaron tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

➤ **Descripción de las disciplinas**

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos, Análisis y diseño en AUP están unidos en la disciplina modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina, en el caso de prueba se desagrega en 3 disciplinas: Pruebas internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI (Capability Maturity Model Integration) -DEV v1.3 para el nivel 2, serian CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto). Para una mayor comprensión se muestra la siguiente tabla 2 (Sánchez, 2015).

Tabla 2: Descripción de las Disciplinas Variación AUP – UCI (Sánchez, 2015)

Disciplinas AUP	Disciplinas Variación AUP-UCI	Objetivos Disciplinas (Variación AUP-UCI)
Modelo	Modelado de negocio	<p>El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes:</p> <p>1- Casos de Uso del Negocio (CUN). 2- Descripción de Proceso de Negocio (DPN). 3- Modelo Conceptual (MC).</p> <p>A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos.</p>
	Requisitos	<p>El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.</p>
	Análisis y diseño	<p>En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.</p>

Implementación	Implementación	Es la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.
Prueba	Pruebas internas	En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas.
	Pruebas de liberación	Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
	Pruebas de Aceptación	Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.
Gestión de configuración	Se cubren con las áreas de procesos PM, PMC y CM que propone CMMI-DEV v1.3. Las mismas son áreas de procesos de gestión y soporte respectivamente.	Consultar en mejoras.prod.uci.cu los libros de procesos de cada una de estas áreas.
Gestión de proyecto		
Entorno		

➤ **Escenarios para la disciplina Requisitos.**

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma:

- Escenario 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

- Escenario 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

➤ **Características por escenarios**

Escenario 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los CUN muestran como los procesos son llevados a cabo por personas y los activos de la organización.

Escenario 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Escenario 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Escenario 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información.

Se seleccionó el Escenario 2, ya que el proyecto modela el negocio usando MC. Para diseñar los esquemas del software, trabajando con esta metodología se utiliza el Lenguaje Unificado de Modelado.

1.5.3 Lenguaje de Modelado

“El Lenguaje de Modelado permite describir los procesos que se emplean en el desarrollo del presente trabajo, así como hacer referencia a detalles de los artefactos que se irán creando. Es la notación principalmente gráfica que usan los métodos para expresar un diseño. El modelado permite especificar el

comportamiento deseado en la construcción de un sistema, comprender mejor lo que se construye y se descubren oportunidades de simplificación y reutilización” (Booch, 2004).

UML es la representación mediante diagramas de las funcionalidades del negocio. El uso de lenguajes visuales trae consigo el fácil entendimiento y asimilación por parte del equipo de trabajo, minimizando el tiempo de implementación del producto.

1.5.4 Herramientas CASE

“Las herramientas CASE con un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Mediante el uso de estas herramientas se reduce el tiempo, costo del desarrollo y mantenimiento del software, así como mejorar su calidad” (Pressman, 2005).

Visual Paradigm for UM 8.0

Visual Paradigm es una herramienta CASE, que propicia un conjunto de ayudas para el desarrollo de programas informáticos dando soporte al modelado visual con UML desde la planificación, pasando por el análisis y diseño, hasta la generación del código fuente de los programas y la documentación. Es soportada por varios sistemas operativos tales como: Ubuntu, Windows y otros. Esta herramienta ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de varios tipos de diagramas. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos (Paradigm, 2008).

Visual Paradigm ofrece:

- Entorno de creación de modelos conformes a UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Disponibilidad en múltiples plataformas.
- Disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado (IDE).
- Disponibilidad de múltiples versiones, para cada necesidad.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Extensible mediante desarrollo de nuevos Módulos (plugins).

Visual Paradigm es la herramienta seleccionada para el modelado UML por las múltiples ventajas que presenta. Además, de ser la herramienta utilizada por el equipo de trabajo del proyecto Aplicativo-SIG,

permite el entendimiento del equipo de desarrollo así como la uniformidad en la documentación asociada a la Plataforma GeneSIG.

1.5.5 Servidor Web

Un servidor web es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario. El servidor web es el encargado de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo en correspondencia con los comandos solicitados (textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música) (Asenjo, 2012).

Servidor de Aplicaciones. Apache

El servidor Apache permite compartir información desde una computadora personal hacia Internet y empleado de manera local se utiliza para la prueba y visualización de la implementación del sistema. La principal ventaja que proporciona es que es de código abierto para plataformas Unix (BSD, GNU/Linux, entre otros), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1.

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección Core y diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web. Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache o que utilizarán características propias de este servidor web (Asenjo, 2012).

1.5.6 Lenguajes de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Programación, 2009).

Lenguajes del lado del cliente

En las aplicaciones basadas en tecnologías web los lenguajes del lado del cliente son aquellos que se interpretan en el navegador del usuario. La selección del navegador a utilizar es independiente de cada usuario, por lo que se debe prestar especial cuidado en la utilización de estos lenguajes para lograr compatibilidad entre los diferentes estándares que los navegadores utilizan para interpretarlos.

➤ JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como un lenguaje orientado a objetos, basado en prototipos, imperativo y dinámico. Es utilizado principalmente del lado del cliente permitiendo crear efectos atractivos y dinámicos en las páginas web.

“Este lenguaje posee varias características tales como: es un lenguaje simple, no hace falta tener conocimientos avanzados de programación para poder hacer un programa en él. Está basado en acciones que posee menos restricciones. Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros” (Valdés, 2007). Además, permite hacer validaciones del lado del cliente permitiendo esto un menor tiempo de respuesta y evita efectuar llamadas incorrectas al servidor.

➤ ExtJS 4.1 librería de Javascript

ExtJS es una librería Javascript de alto rendimiento que permite construir aplicaciones complejas haciendo uso de tecnologías como: AJAX, DHTML y DOM Internet. Permite realizar completas interfaces de usuario, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio (Rosas, 2008). Esta librería incluye:

- Componentes de interfaz de usuario de alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API (Application Programming Interface) fácil de usar.
- Licencias de códigos abiertos y comerciales.

ExtJS permite crear Aplicaciones Ricas en Internet (RIA, por sus siglas en inglés) mediante JavaScript, lo que significa que estas se ejecutan en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, aumentando la interactividad, velocidad y usabilidad en las aplicaciones (Rodríguez, y otros, 2009).

Lenguajes del lado del servidor

Son lenguajes ejecutados en el servidor web. Este es el encargado de recibir una petición hecha por el usuario, procesarla y enviar una respuesta.

➤ **PHP 5.3 (Hypertext Preprocessor)**

PHP es un lenguaje de programación interpretado que se utiliza para la generación de páginas web de forma dinámica. Es un lenguaje de código abierto, gratuito y multiplataforma lo que quiere decir que puede ser ejecutado en la mayoría de los sistemas operativos tales como Unix y Windows. Este lenguaje es usado principalmente en interpretación del lado del servidor, pero actualmente puede ser usado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas de QT y GTK+ (Cowburn, 2016).

Entre sus características fundamentales se encuentran:

- Implementa el paradigma orientado a objeto mediante la utilización de clases las cuales poseen propiedades y métodos y pueden ser instanciadas.
- El código PHP, el cual generalmente define la lógica del funcionamiento interno de un sistema, es invisible al navegador web pues el intérprete PHP es encargado de ejecutarlo y enviar solo el resultado HTML al cliente, esto hace que la programación sea segura y confiable.
- Posee capacidad de conexión con varios gestores de bases de datos como son MySQL y PostgreSQL.
- Es ampliamente extensible mediante la instalación de extensiones que son desarrolladas por la amplia comunidad que brinda soporte.
- Posee detallada documentación, pública desde su sitio web oficial y que está disponible, al igual que el lenguaje y los intérpretes, bajo la licencia libre.

➤ **PHP/ MapScript**

Las librerías MapScript de PHP constituyen la vía de comunicación de las aplicaciones SIG con el servidor de mapas MapServer y por otra parte rompen en cierta medida ésta rigidez de la representación de mapas a través de los ficheros "Mymapfile.map" permitiendo modificar el mapfile en tiempo de ejecución, a los cuales se les conoce como mapfile dinámicos, facilitando de esta forma la creación de aplicaciones con un grado de personalización mayor, eventualmente no alcanzado con aplicaciones del MapServer en modo CGI (Morissette, y otros, 2016).

Para la implementación del sistema de información geográfica se requiere del empleo de programas para almacenar los datos, definir las propiedades de visualización de estos y la elaboración de la interfaz. Para ello es necesario crear una BD, que en el caso de los SIG posee características particulares debido a la

componente espacial que presentan los datos. A continuación se describen el sistema gestor de base de datos usado.

1.5.7 Gestores de bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permiten definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Avila, 2014).

➤ PostgreSQL 9.1 como SGBD

PostgreSQL es un Sistema Gestor de Base de Datos Relacionales Orienta a Objetos (ORDBMS por sus siglas en inglés). Su código fuente está disponible libremente, publicado bajo la licencia BSD (Berkeley Software Distribution). Posee una amplia variedad de plataformas tales como: Linux, Windows, Solaris, Mac OS X y otros. Soporta la realización de transacciones seguras, vistas, uniones, claves extranjeras, procedimientos almacenados, triggers, tipos de datos definidos por el usuario y otras operaciones que se realizan en ellos. El almacenamiento es confiable y consistente. La manipulación es potente, flexible y eficiente. Está considerado como uno de los gestores de bases de datos de código abierto más avanzado en la actualidad (Guerrero, 2010).

Se decidió utilizar este sistema gestor de base de datos para trabajar en el desarrollo del sistema por las características con las que cuenta el mismo, además es el sistema gestor de base de datos utilizado por la plataforma GeneSIG. Este sistema usa como extensión geográfica PostGIS, para añadir soporte a objetos geográficos.

➤ PostGIS 1.5 extensión geográfica de PostgreSQL

Con la finalidad de que la base de datos PostgreSQL soporte objetos geográficos se ha desarrollado la extensión geográfica PostGIS, convirtiéndola en una base de datos espacial que se puede utilizar en SIG. Está liberado bajo la licencia pública general de GNU (GPLv2). Añade soporte para objetos geográficos que permite consultas de ubicación para ser ejecutadas en SQL y es compatible con el servidor de mapas MapServer (PostGIS, 2015).

1.5.8 Entornos de desarrollo integrados (IDE)

Un IDE es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. El IDE de desarrollo seleccionado para la implementación del sistema es NetBeans 7.3, a continuación se definen las características del mismo.

➤ **NetBeans IDE 7.3**

NetBeans IDE es un entorno de desarrollo, es una herramienta de código abierto bajo la licencia pública (GPL). Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones web, o para dispositivos móviles. Da soporte a varias tecnologías como Java, PHP, Groovy, C/C++, HTML5. Entre sus funcionalidades permite escribir, depurar, compilar y ejecutar programas. Además, se destacan sus ventajas de auto completamiento de código, interfaz para el diseño de GUI y la capacidad para importar clases (Hernandez, 2014).

NetBeans fue elegido como IDE debido a que simplifica las tareas de programación, permitiendo centrarse en las particularidades de la aplicación a desarrollar y evitando la complejidad inherente a cualquier desarrollo sobre una plataforma gráfica. Presenta gran área de edición donde el desarrollador interactúa con el código fuente, panel que permite la gestión de los archivos del proyecto y panel de visualización de los mensajes de error, resultado de una compilación.

1.5.9 Servidor de mapas. MapServer 6.0

Para la visualización de los datos se trabaja con el servidor de mapas MapServer. Este servidor es un motor de procesamiento de datos geográficos, de código abierto y creado en C. Permite crear “mapas de imágenes geográficos”, es decir, los mapas que pueden dirigir contenido al usuario. Tiene soporte para varios lenguajes de programación como: PHP, Java, Python, entre otros y es multiplataforma (MapServer, 2015).

El funcionamiento con este programa es sencillo, se define un archivo de texto con extensión .map en el cual se especifican las propiedades y características del mapa que se visualiza, los datos del mismo se organizan por capas que fueron divididos en clases, permitiendo que cada una de ellas se observe con diferentes estilos.

MapServer posee entre sus principales características:

- Una salida cartográfica avanzada que posibilita ejecutar y dibujar elementos según la escala, así como automatizar los componentes del mapa como son la leyenda, la barra de escala y el mapa de referencia.
- Brinda soporte a los lenguajes de script y plataformas más reconocidas, también a formatos vectoriales como PostGIS y formatos ráster como JPG, PNG, GIF, entre otros.
- Es compatible con múltiples sistemas operativos.

Se utilizó este servidor de mapas por disponer de un poderoso sistema cartográfico, por contar con una multitud de formatos ráster y vectorial soportados, por ser rápido y ligero. Además se tiene en cuenta que la plataforma GeneSIG también utiliza este servidor MapServer.

1.6 Conclusiones del capítulo

En este capítulo, se realizó un análisis de los principales conceptos relacionados con el objeto de estudio para tener un mejor entendimiento del mismo. Se realizaron entrevistas a varios trabajadores del MINAG y campesinos independientes, donde se identificó que el sistema agrícola cubano presenta problemas relacionados con la planificación de sus procesos. La ausencia de sistemas informáticos obliga a gestionar la información en formato duro, ocasionando pérdidas de estos registros en muchas ocasiones. Se llevó a cabo un estudio de los principales sistemas similares al que se desea implementar, haciendo énfasis en sus características, ventajas, desventajas y funcionalidades que implementan. Estos sistemas a pesar de ser software privativo, aportan funciones que pueden ser incluidos en el desarrollo del sistema como: la gestión de cultivos, parcelas, transporte, entre otros.

Capítulo 2: Análisis y diseño del SIG para la Gestión de Cultivos

2.1 Introducción

En este capítulo se realiza una descripción del entorno donde se va a desarrollar el sistema por medio del modelo del dominio, donde se representan las clases más significativas del sistema. Se especifican los requisitos funcionales y no funcionales con los que debe cumplir el mismo. La solución propuesta se modela a través de los diagramas de clases del diseño. Se define la arquitectura, la cual es necesaria para comprender el sistema, organizar el desarrollo del mismo, fomentar la reutilización y controlar la evolución del proyecto y el estilo arquitectónico Llamada y Retorno que es el que se emplea en este trabajo. También se definieron los patrones de diseño que son necesarios para garantizar un correcto funcionamiento del sistema. Además, se representa el diagrama de la base de datos que se usó para el desarrollo de la aplicación.

2.2 Modelo del dominio

El modelo de dominio representa las clases más significativas del sistema. El mismo describe los tipos o eventos del entorno en que se desarrolla el sistema, es el artefacto más importante que se crea durante el análisis orientado a objetos. Utilizando la notación UML, representa los objetos del dominio o clases conceptuales, las asociadas entre ellas y sus atributos (Larman, 2003).

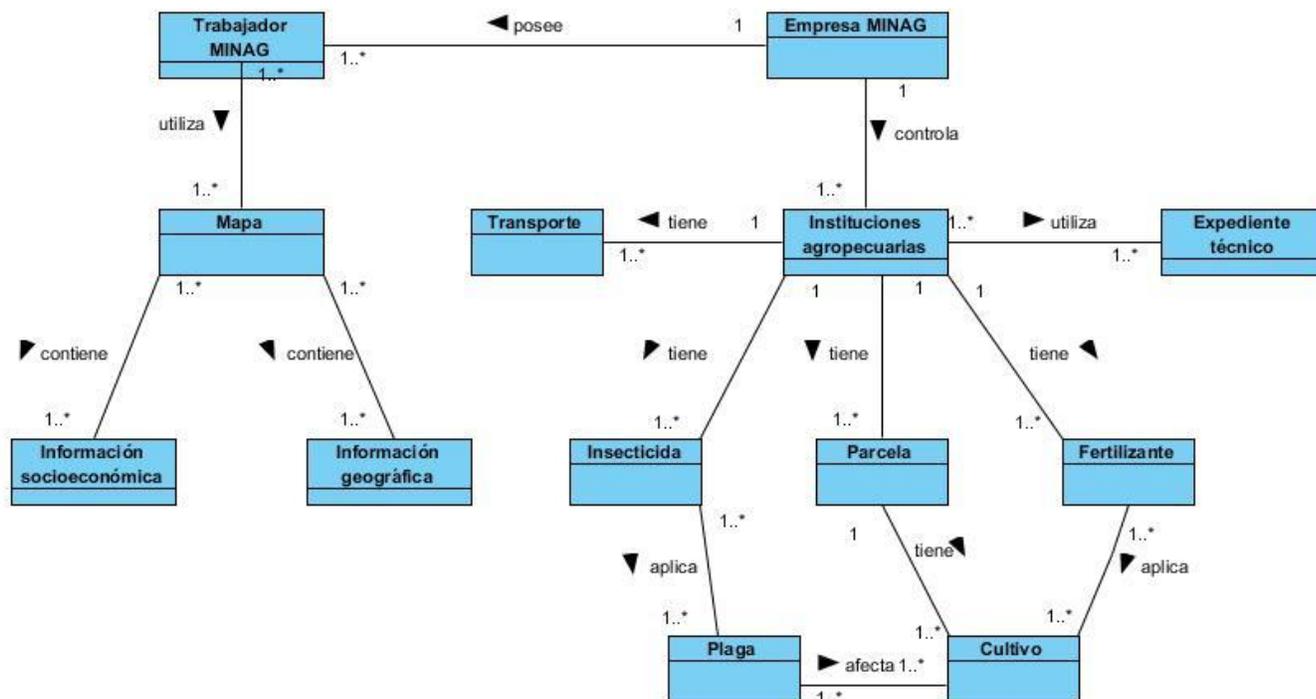


Fig 6: Modelo de dominio

Tabla 3: Descripción de las clases del modelo del dominio

Concepto	Descripción
MINAG	Ministerio de la Agricultura de Cuba, es la institución cubana encargada de lograr el desarrollo sostenible de las producciones agropecuarias y forestales del país con destino a la satisfacción de las necesidades alimentarias de la población.
Trabajadores_MINAG	Persona que trabaja en alguna de las instalaciones del Ministerio de la Agricultura de Cuba y que necesite consultar algún tipo de información.
Mapa	Es la representación gráfica de una porción de territorio; tiene propiedades métricas y permite medir distancias, ángulos y otros.
Información Socioeconómica	Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés.
Información Geográfica	Conjunto organizado de datos espaciales georreferenciados.
Instituciones Agropecuarias	Instituciones cubanas destinadas a la agricultura.
Transporte	Vehículo que se usa para el desplazamiento de personas, animales u objetos.
Parcela	Espacio de tierra que se va a cultivar.
Insecticida	Compuesto químico que se usa para tratar plagas.
Fertilizante	Sustancia orgánica para mejorar el crecimiento y calidad de las plantas.
Expediente Técnico	Documento que almacena toda la información referente a los cultivos.
Plaga	Organismo vivo que afecta los cultivos.
Cultivo	Producto Agrícola.

2.3 Especificación de los requisitos del sistema

Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. Estos pueden clasificarse en requisitos funcionales y no funcionales. Los requisitos funcionales definen qué debe hacer un sistema y los requisitos no funcionales definen cómo debe ser el sistema (Sommerville, 2005).

2.3.1 Requisitos funcionales (RF) del sistema

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, para que las peticiones del cliente queden satisfechas (Sommerville, 2005).

RF1 Insertar parcela: este requisito permite que el usuario pueda insertar una parcela con los siguientes parámetros: el nombre de la parcela, el tipo de suelo, si el suelo es húmedo o no y el área que es calculable, este valor se obtiene cuando se dibuja la parcela en el mapa.

RF2 Modificar parcela: este requisito permite que el usuario pueda modificar los parámetros de una parcela determinada como: el nombre, el área, el tipo de suelo y si el suelo es húmedo o no.

RF3 Eliminar parcela: este requisito permite que el usuario pueda eliminar una parcela.

RF4 Insertar cultivo: este requisito permite que el usuario pueda insertar un cultivo con los siguientes parámetros: el nombre del cultivo, tiempo de germinación, tiempo de cosecha, tiempo entre fertilizantes, tiempo entre riego, la estación del año en que se encuentre y el precio del cultivo que se va a insertar dado en (kg, toneladas, Lb y en Unidades).

RF5 Modificar cultivo: este requisito permite que el usuario pueda modificar los parámetros de un cultivo determinado como: nombre del cultivo, tiempo de germinación, tiempo de cosecha, tiempo entre fertilizantes, tiempo entre riego, la estación del año en que se encuentre y el precio del cultivo que se va a insertar, está dado en (kg, toneladas, Lb y Unidades).

RF6 Eliminar cultivo: este requisito permite que el usuario pueda eliminar un cultivo.

RF7 Insertar alerta: el sistema debe permitir a los usuarios insertar una nueva alerta con los siguientes parámetros: el nombre de la alerta, el tipo de alerta (cosechar, fertilizar, regar y fumigar), una breve descripción de la misma, el estado de la alerta (activado o desactivado) y el tiempo de antelación que se define para que se muestre la alerta.

RF8 Modificar alerta: el sistema debe permitir a los usuarios modificar los parámetros de las alertas tales como: nombre de la alerta, el tipo de alerta (cosechar, fertilizar, regar y fumigar), una breve descripción de la misma, el estado de la alerta (activado o desactivado) y el tiempo de antelación que se define para que salga la alerta.

RF9 Eliminar alerta: este requisito permite que el usuario pueda eliminar una alerta.

RF10 Insertar plaga: este requisito permite que el usuario pueda insertar la plaga determinada con los siguientes parámetros: el nombre y el tipo de plaga.

RF11 Modificar plaga: este requisito permite que el usuario pueda modificar los parámetros de una plaga tales como nombre y el tipo de plaga.

RF12 Eliminar plaga: este requisito permite que el usuario pueda eliminar una plaga.

RF13 Insertar fertilizante: este requisito permite que el usuario pueda insertar un fertilizante con los siguientes parámetros: el nombre y el tipo de fertilizante.

RF14 Modificar fertilizante: este requisito permite que el usuario pueda modificar los parámetros de un fertilizante tales como el nombre y el tipo de fertilizante.

RF15 Eliminar fertilizante: este requisito permite que el usuario pueda eliminar un fertilizante.

RF16 Insertar transporte: este requisito permite que el usuario pueda insertar el medio de transporte en el que se trasladarán los cultivos una vez concluidas sus fases de recogida, del mismo se conoce el tipo de transporte, la capacidad y si está disponible o no.

RF17 Modificar transporte: este requisito permite que el usuario pueda modificar los parámetros de un transporte tales como el tipo de transporte, la capacidad y la disponibilidad.

RF18 Eliminar transporte: este requisito permite que el usuario pueda eliminar un transporte.

RF19 Insertar insecticida: este requisito permite que el usuario pueda insertar un insecticida con los siguientes parámetros: el nombre y el tipo de insecticida.

RF20 Modificar insecticida: este requisito permite que el usuario pueda modificar los parámetros de un insecticida tales como el nombre y el tipo de insecticida.

RF21 Eliminar insecticida: este requisito permite que el usuario pueda eliminar un insecticida.

RF22 Mostrar cultivos a menos de 10 días de recogida: este requisito permite que el usuario pueda consultar los cultivos que le falten menos de 10 días para su recogida. El usuario puede consultar esta información en la funcionalidad mostrar alerta.

RF23 Mostrar tierras ociosas: este requisito permite mostrar en el mapa todas las parcelas que estén sin sembrar en ese momento.

RF24 Mostrar cultivos que se afectan por una plaga determinada: este requisito permite representar en el mapa los cultivos afectados por una plaga determinada y los cultivos cercanos que tienen el mismo producto que pueden ser afectados por la misma plaga. Se tiene en cuenta para realizar esta recomendación una distancia de 3 km, el cual se va a representar con otro color.

RF25 Mostrar alerta: en este requisito se muestran todas las alertas del sistema que estén activas. En la descripción de cada alerta se tiene en cuenta el nombre, el tipo de alerta (cosechar, fumigar, fertilizar y regar) y el tiempo.

Módulo de Navegación

RF26 Acercar una región del mapa: Permite aumentar el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de acercar.

RF27 Alejar una región del mapa: Permite disminuir el tamaño del mapa (en caso de que el mapa no se muestre en la vista principal), mostrando en pantalla la región donde el usuario realice la operación de alejar.

RF28 Visualizar todo el mapa: Permite visualizar el mapa según la vista inicial de la aplicación (en caso de que el mapa no se muestre en la vista principal).

RF29 Modificar el centro del mapa: Permite modificar el centro del mapa (en caso de que el mapa no se muestre en la vista principal), sin cambiar la escala del mismo, poniendo como centro el nuevo punto seleccionado por el usuario.

RF30 Mover el mapa: Permite al usuario mover el mapa para visualizar otras regiones (en caso de que el mapa no se muestre en la vista principal).

RF31 Navegar utilizando el mapa de referencia: Permite 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. En el mapa de referencia se marca con un rectángulo de bordes rojos la región que se visualiza en el mapa principal. Si la escala en que se muestra el mapa es solo se mostrará en el Mapa de Referencia una cruz de color rojo para identificar el lugar que se visualiza en el mapa principal.

RF32 Visualizar el mapa anterior: Permite visualizar el mapa anterior al que se visualiza en la aplicación (en caso de que el usuario haya realizado algún cambio en la visualización del mapa).

RF33 Visualizar el mapa siguiente: Permite que, una vez que haya seleccionado la opción descrita en el RF26, la aplicación visualice nuevamente el mapa que se encontraba en pantalla.

Módulo de Identificación

RF34 Realizar identificación puntual: Permite al usuario consultar la información asociada a los objetos representados sobre el mapa. El usuario puede seleccionar qué elementos desea identificar y, una vez que marque el punto sobre el mapa, se le mostrará la información de todos los elementos encontrados en correspondencia con los definidos por él con anterioridad. En caso de existir un orden de prioridad para

dichos elementos, el usuario solo debe marcar el punto deseado sobre el mapa y se le mostrará la información del elemento de mayor prioridad encontrado en esa ubicación.

RF35 Realizar identificación rectangular: Permite al usuario consultar la información asociada a los objetos representados sobre el mapa. El usuario debe seleccionar qué elementos desea identificar y, una vez que marque el rectángulo sobre el mapa, se le mostrará la información de todos los elementos encontrados dentro del rectángulo dibujado y en correspondencia con los definidos por él con anterioridad.

RF36 Realizar identificación poligonal: Permite al usuario consultar la información asociada a los objetos representados sobre el mapa. El usuario debe seleccionar qué elementos desea identificar y, una vez que marque el polígono sobre el mapa, se le mostrará la información de todos los elementos encontrados dentro del polígono dibujado y en correspondencia con los definidos por él con anterioridad.

RF37 Realizar identificación circular: Permite al usuario consultar la información asociada a los objetos representados sobre el mapa. El usuario debe seleccionar qué elementos desea identificar y, una vez que marque el círculo sobre el mapa, se le mostrará la información de todos los elementos encontrados dentro del círculo dibujado y en correspondencia con los definidos por él con anterioridad.

Módulo de Localización

RF38 Buscar elemento a localizar: Permite al usuario buscar el elemento deseado para localizarlo sobre el mapa posteriormente. Para ello debe especificar el elemento que desea, así como la región y el atributo correspondiente a dicho elemento. Como resultado se muestra un listado con los elementos encontrados que se correspondan con las especificaciones dadas por el usuario.

RF39 Localizar por elemento: Permite al usuario localizar sobre el mapa el elemento deseado. Para ello debe especificar dicho elemento y, como resultado, se mostrará el mismo marcado sobre el mapa con una tachuela. El mapa se muestra de forma tal que sea posible distinguir el elemento localizado.

RF40 Localizar por coordenadas: Permite al usuario localizar sobre el mapa los elementos que se encuentren en las coordenadas indicadas por este. Como resultado se mostrará sobre el mapa una tachuela indicando el lugar correspondiente a las coordenadas especificadas. El mapa se muestra de forma tal que sea posible distinguir dicho lugar.

Módulo de análisis

RF41 Realizar tematización por colores: Esta funcionalidad permite que el usuario pueda crear un mapa temático por colores y mostrar esta tematización en el mapa en función del criterio de análisis que haya seleccionado. Esta funcionalidad requiere los siguientes criterios de entrada:

- Criterio de análisis (Formato: Lista desplegable, Obligatorio: Sí).

Una vez definidos los rangos, estos se mostrarán con el color con que se visualizarán en el mapa, pero el usuario podrá modificar este color para cada rango en caso que lo desee. El sistema mostrará el mapa tematizado por colores y mostrará además los valores numéricos asociados al criterio de análisis de la tematización.

Módulo de reportes

RF42 Estimación de Ingreso: Este requisito permite conocer el estimado de productividad de un cultivo determinado. De este requisito se conocen los siguientes datos: el nombre de la parcela, los cultivos sembrados en esa parcela, la recogida estimada por metro cuadrado que está dado en (Lb, Kg, Unidades, Toneladas) y se convierte a metro, el precio unitario que está dado en (Lb, Kg, Unidades, Toneladas), el área sembrada que está dado en (Ha, m) y el ingreso estimado que está dado en pesos y se calcula de la siguiente forma:

Ingreso estimado = precio unitario * unidad llevada a Kg o unidad * Área sembrada llevada a metros

Nota: Al ingreso total se le resta un 10 % por concepto de pérdidas.

RF43 Recomendación de transporte: Este requisito muestra el transporte indicado para la recogida de un cultivo determinado según la capacidad necesaria para su transportación. Del mismo se conocen los siguientes parámetros: la parcela, los cultivos sembrados en esta parcela, la cantidad de cultivo que se quiere transportar, según esta cantidad se recomienda el transporte más indicado para realizar esta acción, se conoce la capacidad del transporte recomendado y las unidades (la cantidad de transporte que hace falta según la capacidad del mismo y la cantidad de cultivo a transportar)

RF44 Recomendar insecticida para una plaga determinada: Este requisito muestra el insecticida indicado para erradicar una plaga que esté atacando un cultivo determinado. Del mismo se conoce el cultivo afectado por la plaga, el nombre de la plaga detectada en el cultivo, el nombre del insecticida que se recomienda para eliminar esta plaga y el tipo de insecticida al que pertenece.

RF45 Recomendar fertilizante para un cultivo determinado: Este requisito muestra el fertilizante indicado para tratar un cultivo determinado. Del mismo se conocen la siguiente información: el nombre del cultivo, el nombre del fertilizante que se recomienda aplicar al cultivo y el tipo de fertilizante.

2.3.2 Requisitos no funcionales (RNF) del sistema

Los requisitos no funcionales, son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta

y la capacidad de almacenamiento. De forma alternativa, define las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del mismo (Sommerville, 2005).

RNF1 De Usabilidad

El sistema podrá ser utilizado por cualquier usuario con conocimientos básicos de computación y SIG. La solución propuesta usa iconos para facilitar el reconocimiento de las funcionalidades por parte de los usuarios.

RNF2 Interfaz de usuario

El sistema debe tener un diseño sencillo, donde no sea necesario mucho entrenamiento para utilizar el sistema.

RNF3 De Hardware

Para las computadoras clientes:

- 512 MB de RAM como mínimo.
- Procesador 512 MHz como mínimo.

Para los servidores:

- El Servidor de Mapas debe tener como mínimo 2 GB de RAM.
- El Servidor de base de datos debe tener como mínimo 2 GB de RAM.
- Procesador 3 GHz como mínimo.

RNF4 De Software

Para las computadoras cliente:

- Se recomienda que esté instalado el navegador web Mozilla Firefox.
- Sistema operativo: GNU/Linux, Windows 7 o superior.

Para los Servidores:

- Sistema operativo GNU/Linux Ubuntu Server 12.04.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.

- MapServer 6.0, con extensión PHP MapScript.

RNF5 Restricciones de licencia

El sistema puede ser utilizado, modificado y distribuido, sin necesidad de obtener la autorización de sus respectivos titulares.

RNF6 Requisitos legales, de Derecho de autor y otros

El sistema debe cumplir con las leyes, decretos, resoluciones y manuales establecidos por la República de Cuba y la Universidad.

2.4 Arquitectura del sistema

La arquitectura es una vista estructural de alto nivel, que define estilo o combinación de estilos para una solución. Se puede decir que la arquitectura es esencial para éxito o fracaso de un proyecto. Además, la arquitectura de un software es necesaria para comprender el sistema, organizar el desarrollo del mismo, fomentar la reutilización y controlar la evolución del proyecto (Pressman, 2005). GeneSIG tiene una estructura basada en el *framework CartoWeb* que posee una arquitectura cliente – servidor. La plataforma cumple entonces con la estructura definida por CartoWeb, pero utiliza solo los paquetes a los cuales les realizara cambios o aportes funcionales.

2.4.1 Patrones arquitectónicos

“Un patrón arquitectónico provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Estos patrones son plantillas para arquitecturas de software concretas, que especificas las propiedades estructurales de una ampliación con amplitud de todo el sistema y tiene un impacto en la arquitectura de subsistemas” (Camacho, y otros, 2004)

La selección de los patrones arquitectónicos es una decisión fundamental en el desarrollo de un software, pues permiten darle solución a un problema en específico. En la solución de la investigación se usa la Arquitectura Orientada a Objetos y la Arquitectura Basada en Componentes sobre las que se basa la plataforma GeneSIG.

Arquitectura Orientada a Objetos: nombres alternativos para este estilo han sido Arquitecturas Basadas en Objetos, Abstracción de Datos y Organización Orientada a Objetos. Los componentes del estilo se basan en principios orientados a objetos: encapsulamiento, herencia y polimorfismo. Las interfaces están separadas de las implementaciones. Las representaciones de los datos y las operaciones están

encapsuladas en un tipo abstracto de datos u objeto. La comunicación entre los componentes es a través de mensajes (Reynoso, y otros, 2004).

Arquitectura Basada en Componentes: se centra en el diseño y construcción de sistemas computacionales que utilizan componentes de software reutilizables. Define la composición de software como el proceso de construir aplicaciones mediante la interconexión de componentes de software a través de sus interfaces (de composición), aboga por la utilización de componentes prefabricados sin tener que desarrollarlos de nuevo (Robaina, 2008).

2.4.1 Estilo Arquitectónico

Expresa componentes y las relaciones entre estos, con las restricciones de su aplicación y la composición asociada, así como también las reglas para su construcción. Se consideran como un tipo particular de estructura fundamental para un sistema de software, conjuntamente con un método asociado que especifica cómo construirlo incluyendo información acerca de cuándo usar la arquitectura que describe, sus invariantes y especializaciones, así como las consecuencias de su aplicación (Camacho, y otros, 2004).

Los estilos arquitectónicos son un tipo particular de estructura que definen los posibles patrones a usar en la implementación de la aplicación. Los estilos más conocidos son: Flujo de Datos, Centrado en datos y, Llamada y Retorno que es el que se emplea en este trabajo (Reynoso, y otros, 2004).

Luego de haber analizado la arquitectura de la plataforma GeneSIG y haber decidido mantener la misma para el sistema a desarrollar, se hace necesario entonces definir los patrones de diseño que se pueden emplear en la implementación de la solución.

2.5 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Con el uso de los patrones se facilitan la reusabilidad, extensibilidad y mantenimiento de los sistemas (Pressman, 2011).

Entre los patrones de diseño más utilizados se encuentran los patrones GRASP (General Responsibility Assignment Patterns), que representan los principios básicos de la asignación de responsabilidades a objetos y los patrones GOF (Gang of Four) surgen como una forma indispensable de enfrentarse a la programación, estos se clasifican en creacionales, estructurales y de comportamiento (Larman, 2003).

2.5.1 Patrones GRASP

Creador: Asignarle a una clase la responsabilidad de crear una instancia de otra clase (Larman, 2003). Tiene la responsabilidad de identificar quién debe ser el responsable de la creación de nuevo objeto o clase. El uso de este patrón se puede apreciar en el diseño de la solución cuando se le solicita información a la

clase **ClientGestionar** y esta se encarga de crear instancias de la clase **RequestGestionar**. De igual forma la clase **ServerGestionar** crea instancias de la clase **ResultGestionar**.

Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase (Larman, 2003). Está presente en la clase Main, que controla la petición realizada desde la interfaz para acceder a los datos del fichero que atenderán dicha petición y en la clase PluginManager, que controla la petición realizada desde la interfaz para poder acceder al plugin que atenderá dicha petición.

Bajo acoplamiento: es el encargado de asignar las responsabilidades de forma que las clases se comuniquen con el menor número de clases que sea posible. En el modelo de diseño se muestra que no existen dependencias fuertes entre las clases, permitiendo así que al producirse una modificación en alguna de estas, se tenga la mínima repercusión posible en el resto de las clases y pueda ser reutilizado el código (Larman, 2003). Se asignan las responsabilidades de forma tal que cada clase se comunique con el menor número de clases, minimizando el nivel de dependencia.

Alta cohesión: Se encarga de asignar a las clases responsables que trabajen sobre una misma área de la aplicación sin mucha complejidad. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme (Larman, 2003). En el diseño de la solución existen las clases clientes y las clases servidoras, las cuales se encargan de realizar cada una su función, esto permite que las clases no estén saturadas y que se garantice a través de la asignación de responsabilidades, que el sistema presente alta cohesión en sus clases.

2.5.2 Patrones GOF

Singleton (Instancia única): En el diseño de clases es necesario aplicar la solución de este patrón, que no es más que garantizar el acceso único a una clase mediante una única instancia. Por este medio se puede controlar el acceso a las clases. El patrón Singleton se evidencia al modificar el framework CartoWeb, donde el objetivo del mismo es crear el objeto "mapa" para que no se cree cada vez que se hace un envío en la aplicación.

Command (Acción): Este patrón permite encapsular las peticiones a través de un objeto, lo que permite realizar operaciones como gestionar las acciones de dicho objeto. Se utiliza para la comunicación a través de las interfaces de usuario, específicamente a través de la clase AJAXHelper que es la encargada de comunicar las interfaces con el servidor. Uno de los aspectos más importantes en el sistema son las interfaces gráficas de usuario, ya que el usuario interactúa constantemente con ellas y por eso principalmente se aplica este patrón GOF a la solución.

2.6 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tiene impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es de ese modo utilizada como una entrada fundamental de las actividades de implementación (Booch, y otros, 2000).

2.6.1 Diagrama de clases del diseño

Una clase del diseño es una abstracción de una clase real o construcción similar en la implementación del sistema. El lenguaje que se utiliza en dichas clases es el mismo que se emplea para la implementación del sistema. Se especifican los atributos y las operaciones. Además, se pueden realizar interfaces si tienen sentido para la programación y los métodos tienen correspondencia con las operaciones que fueron utilizadas en la implementación (Larman, 2003).

El diagrama de clases del diseño presentado a continuación está basado en la implementación que se realiza con el framework (CartoWeb), incluyendo específicamente todas las clases contenidas en el framework, y siguiendo la premisa de ajustarse al flujo propio de la solución y no a los detalles transparentes al equipo de desarrollo, se decidió representar únicamente las clases y extensiones web que tuvieran que ver directamente con la construcción de la solución, sin dejar de incluir las partes fundamentales del framework que ayuden a su comprensión.

A continuación se muestran los Diagramas de Clases del Diseño correspondientes a los requisitos funcionales Insertar cultivo e Insertar parcela respectivamente.

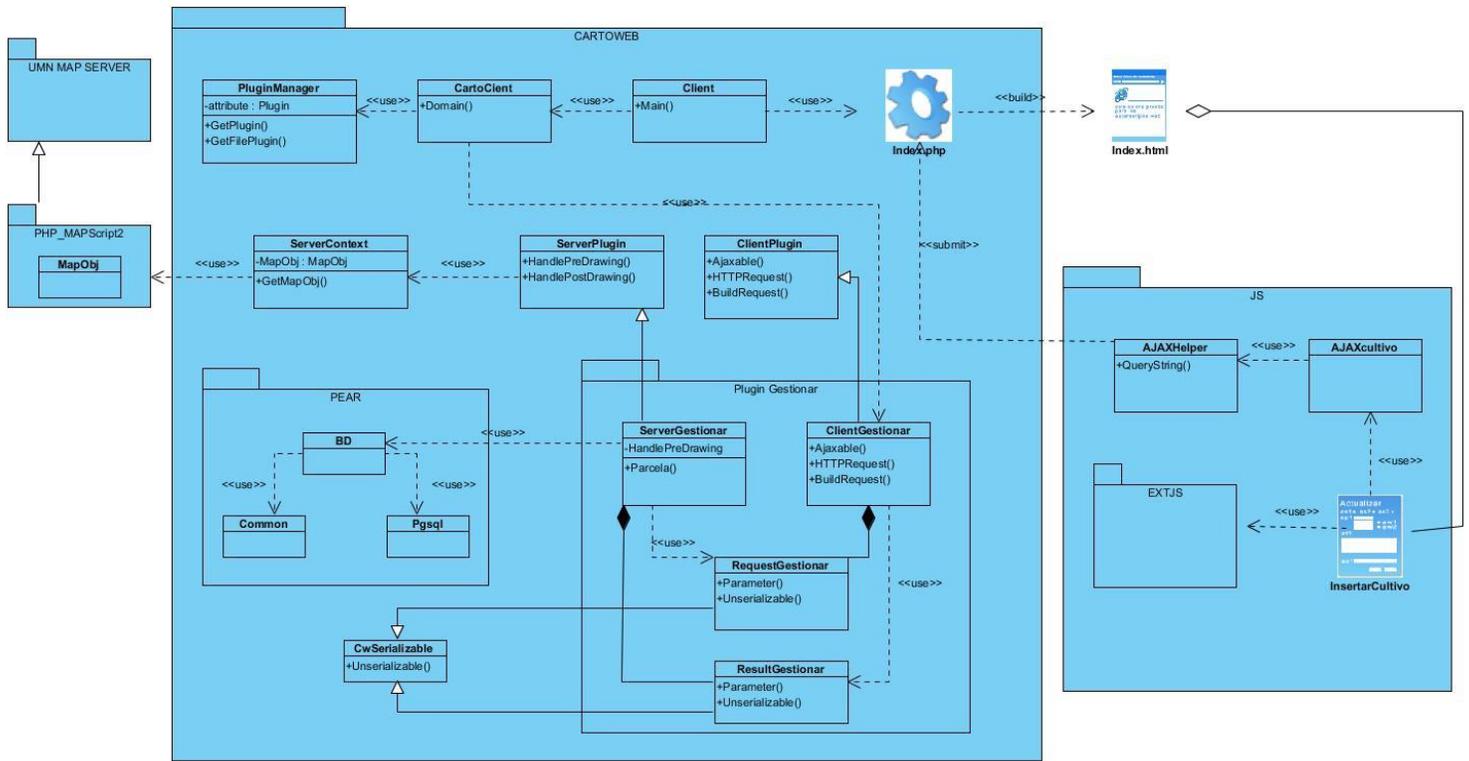


Fig 7: Diagrama de Clases del Diseño Insertar Cultivo

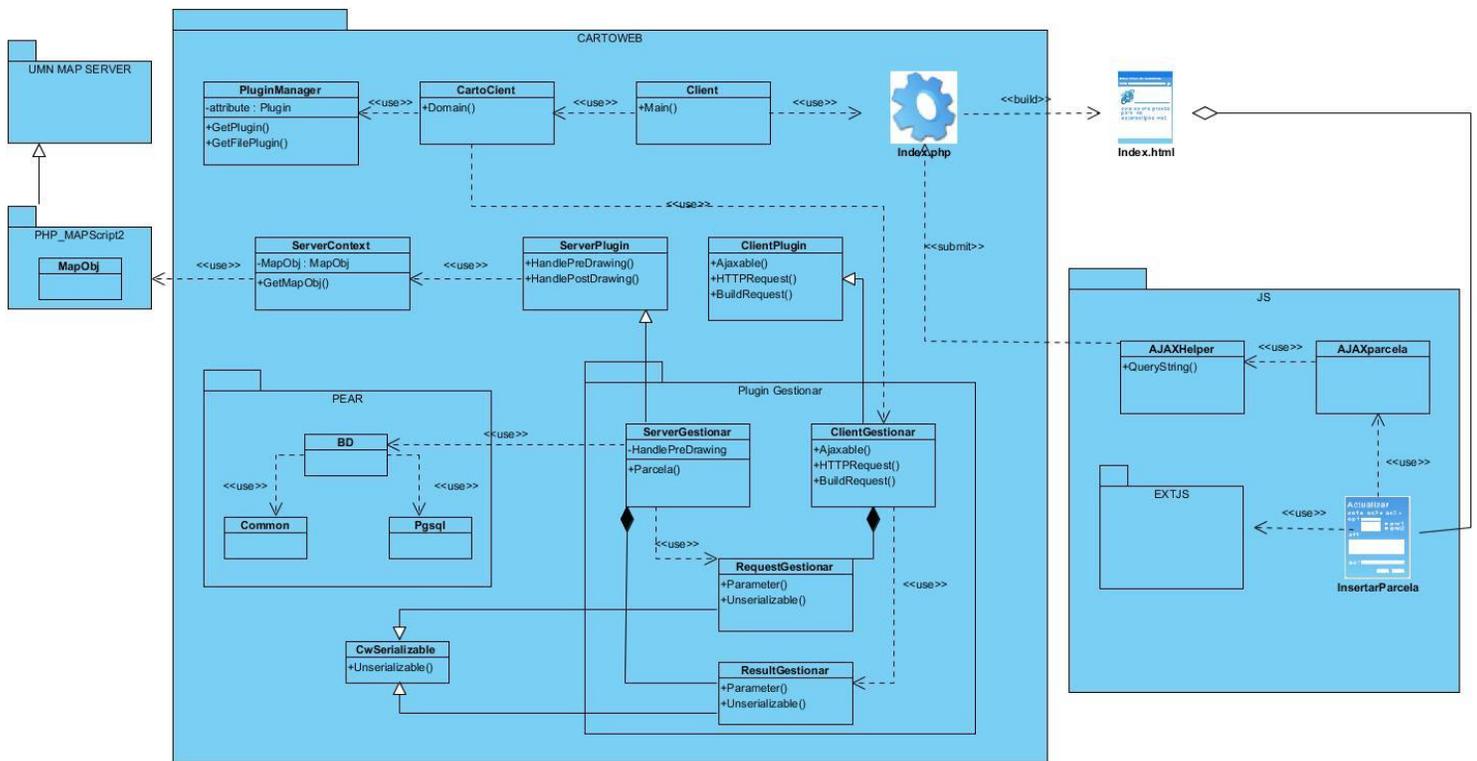


Fig 8: Diagrama de Clases del Diseño Insertar parcela

Para consultar los otros diagramas del diseño puede dirigirse al anexo 2 y a los expedientes del proyecto Modelo de Diseño v1.0, donde se encuentran todos los diagramas de clases del diseño correspondientes a los requisitos funcionales del sistema.

2.6.2 Descripción de las clases del diseño del RF Insertar cultivo

Cada uno de los diagramas de Clases del diseño que emplea el paquete CartoWeb posee clases que son comunes, así como también las clases que son JavaScript y PHP. Para una mejor comprensión del propósito de cada clase del diseño de la solución, se describen las mismas de la Fig 7 Insertar Cultivo:

Tabla 4: Descripción de las clases del diseño

Módulos	Descripción
index.php	Tiene como propósito controlar la realización del RF en sí, recibe las peticiones realizadas por el cliente, gestiona las mismas y manda a construir la ClientPage.
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 RF y se definen una serie de variables globales que van a ser utilizadas por la aplicación.
PluginManager	Clase que se utiliza para gestionar la base de plugins.
ClientPlugin	Contiene las interfaces necesarias para los plugins del lado del cliente.
ServerPlugin	Esta clase 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.
BD	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 la misma satisfactoriamente, esto incluye 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 Client y el Server del plugin.

AJAXHelper	Tiene como propósito enviar las respuestas de los plugins "AJAX", para alimentar a los plugins que responden a las peticiones del usuario.
Index.html	Es la encargada de mostrar en el mapa la parcela seleccionada.
AJAXcultivo	Es la encargada de gestionar el pedido y respuesta a las peticiones del usuario por Ajax.
InsertarCultivo	Permite insertar un cultivo con los siguientes parámetros: el nombre, T.germinación, T. cosecha, T. entre fertilizante, T. entre riego, Estación y el precio del cultivo.
ServerGestionar	Es la clase servidora que tiene como principal función la conexión con la base de datos, permite realizar las consultas requeridas y enviar las respuestas necesarias al ClientGestionar.
ClientGestionar	Se encarga de recoger y seleccionar 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 ServerGestionar.
RequestGestionar	Es una clase común encargada de transportar los datos recogidos en ClientGestionar desde la interfaz y transportarlos a la clase ServerGestionar.
ResultGestionar	Es una clase común encargada de transportar los datos generados en ServerGestionar a la clase ClientGestionar.

2.7 Diseño de la Base de Datos del SIG para la Gestión de Cultivos

El diseño de la BD tiene como principal objetivo asegurarse de que los datos persistentes son almacenados y consistentes. Además, se encarga de definir el comportamiento que debe ser implementado en la base de datos.

2.7.1 Diagrama de clases persistentes

El diagrama de clases persistentes muestra todas las clases capaces de mantener su valor en el espacio y en el tiempo. En la Fig 9 se muestra el diagrama de clases persistentes de la propuesta de solución:

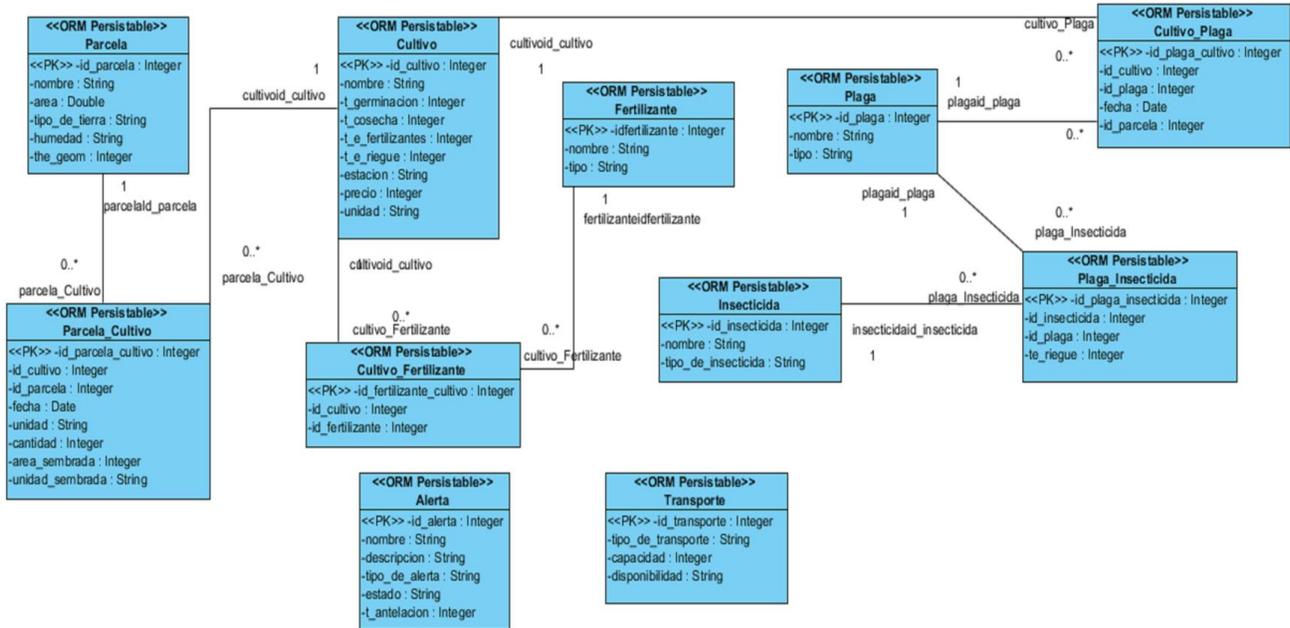


Fig 9: Diagrama de clases persistentes

2.7.2 Diagrama Entidad – Relación

Este diagrama se emplea para identificar un conjunto de componentes primarios como objetos de datos, atributos, relaciones e indicadores de varios tipos, su propósito principal es representar objetos de datos y sus relaciones. En la Fig 10 se muestra el diagrama entidad - relación de la propuesta de solución:

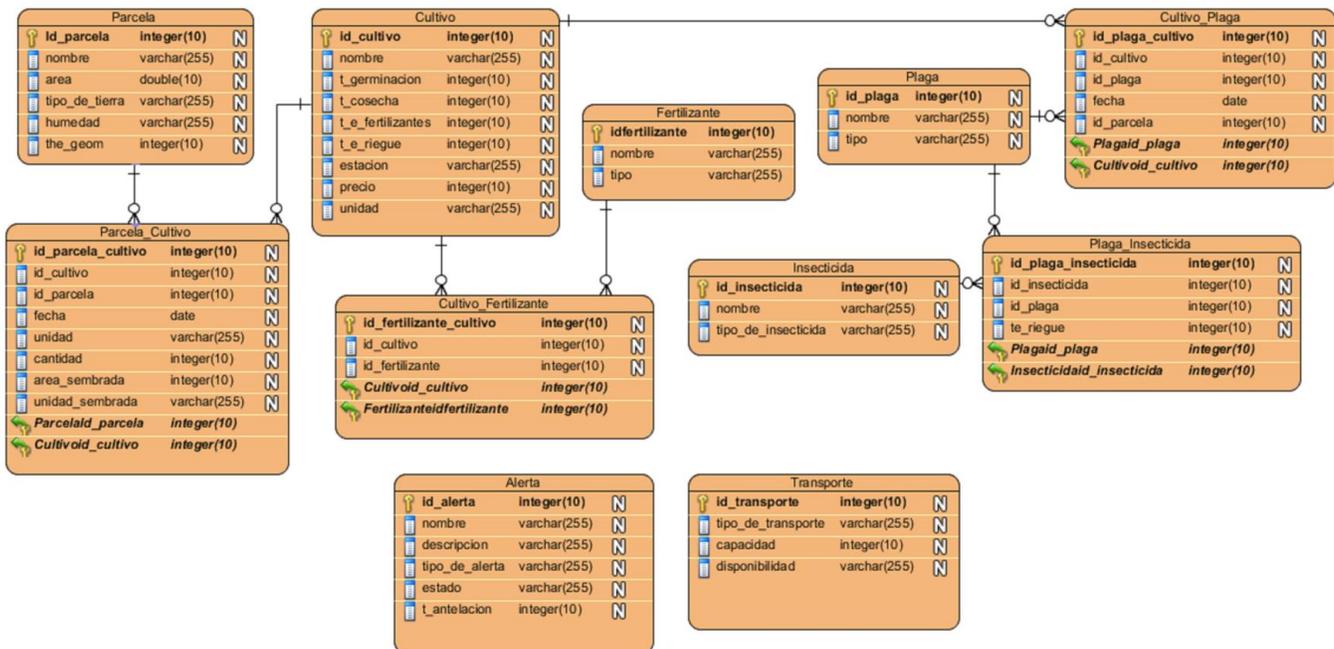


Fig 10: Modelo físico de la base de datos

2.7 Propuesta de solución

La propuesta de solución tiene como objetivo implementar un SIG para la Gestión de Cultivos que mejore la planificación, gestión y asignación de los recursos, la toma de decisiones y el tiempo de respuesta ante situaciones imprevistas. Este sistema permitirá gestionar las parcelas, cultivos, fertilizantes, insecticida, plagas, transporte y alertas generando los siguientes reportes.

- Estimación de Ingreso.
- Recomendación de transporte.
- Recomendar insecticida para una plaga determinada.
- Recomendar fertilizante para un cultivo determinado.

Además permitirá representar geográficamente cada una de las parcelas insertadas en el mapa, y contará con un sistema de generación de alertas que permitirá notificar al usuario con tiempo de antelación las tareas principales que son, regar, cosechar, fumigar, fertilizar los cultivos entre otros.

La aplicación permite tematizar en el mapa las tierras ociosas, los cultivos afectados por una plaga determinada, los cultivos que estén listos para ser recogidos, además de identificar y localizar las parcelas.

2.8 Conclusiones del capítulo

En este capítulo se realiza un análisis mediante el modelo de dominio para comprender la estructura del entorno donde se va a desarrollar el sistema. Se especifican los requisitos funcionales y no funcionales para tener un mayor grado de detalle sobre las características del sistema. Se detallan las categorías de los requisitos del software posibilitando un mejor entendimiento sobre las características a implementar y una línea base arquitectónica robusta. Los artefactos generados durante el capítulo permiten una mejor comprensión del sistema a desarrollar.

Capítulo 3: Implementación del sistema y pruebas

3.1 Introducción

En este capítulo se trata todo lo relacionado a la implementación del sistema y las pruebas realizadas a este con el objetivo de que el producto final cumpla con la calidad requerida para satisfacer las necesidades del cliente y cumplir con sus expectativas. Se realiza el diagrama de componentes para visualizar con mayor facilidad la estructura general del sistema y el diagrama de despliegue para modelar la arquitectura. Además, se estará generando las pruebas aplicadas a la solución informática mediante el método de prueba, caja negra, utilizando la técnica de partición equivalente, así como los resultados que demuestran el grado de satisfacción de la aplicación informática desarrollada.

3.2 Diagrama de componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. En estos diagramas se muestran los elementos de diseño de un sistema de software. Un componente representa una parte de un sistema modular, desplegable, reemplazable, que encapsula la implementación y expone un conjunto de clases (Larman, 2003).

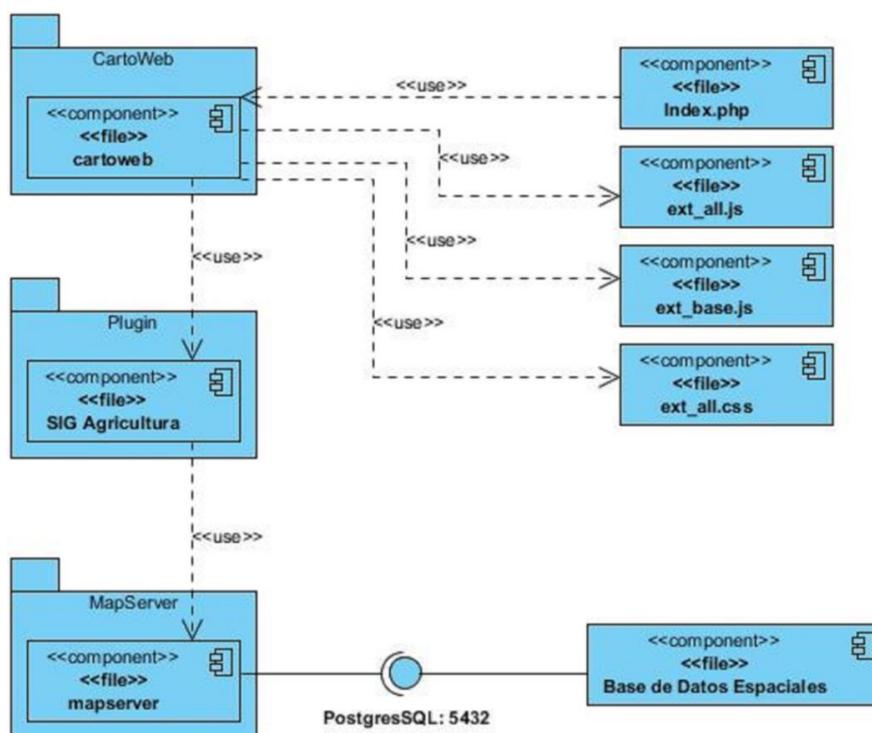


Fig 11: Diagrama de componentes

3.3 Diagrama de despliegue

“Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Las relaciones entre los nodos constituyen medios de comunicación” (UML, 2007).

A continuación en la Fig 12 se muestra el diagrama de despliegue correspondiente al sistema implementado. Se representa un nodo PC_Cliente, el cual se comunica con el nodo Servidor de Aplicación Web y el Servidor de Mapas mediante el protocolo HTTP y a su vez este nodo se comunica con el nodo Servidor BD a través del protocolo TCP IP.

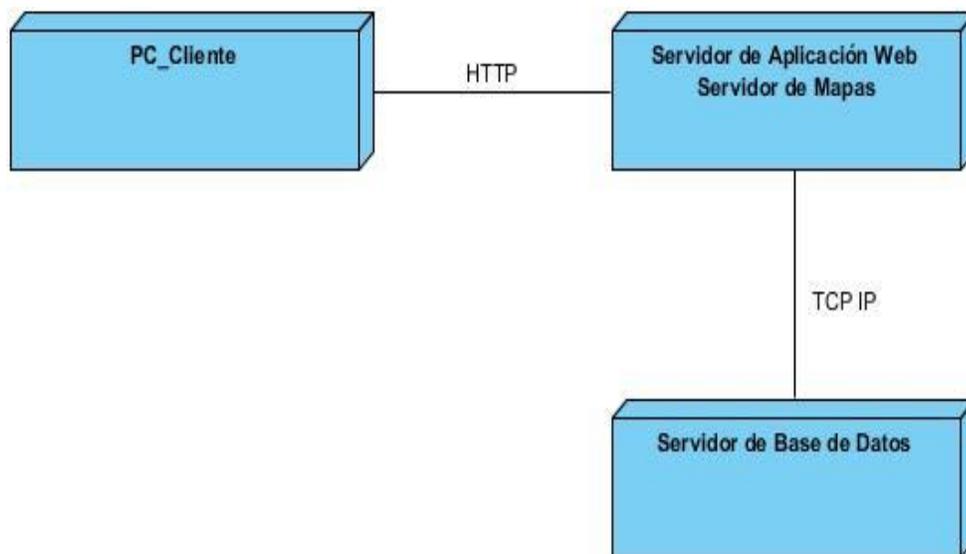


Fig 12: Diagrama de despliegue

3.4 Pruebas del sistema

Las pruebas de software representan actividades claves que ayudan a entregar el producto de forma tal que satisfaga las necesidades del cliente, cumpla con sus expectativas garantizando que el producto cumpla las especificaciones definidas. El objetivo de las pruebas es asegurarse de que los subsistemas de todos los niveles y de todas las capas funcionen. Comenzando por las capas más bajas de la arquitectura se prueban los mecanismos de distribución, almacenamiento, recuperación y concurrencia de objetos, así como otros mecanismos de las capas inferiores del sistema lo que significa que no solo se comprueba la funcionalidad sino también si el rendimiento es aceptable (Booch, y otros, 2000).

3.4.1 Diseño de Casos de Prueba

Un diseño de caso de prueba (DCP) está compuesto por un conjunto de entradas, respuesta que emite el sistema de acuerdo a esas entradas y el flujo central que indica el camino del escenario descrito. Estos son desarrollados para verificar el cumplimiento total o parcial de un requisito (Pressman, 2005). Las entradas representan las variables que se pueden especificar y las mismas contienen: V, I, o N/A. La variable V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Pruebas de caja negra

Se seleccionó en esta etapa para aplicarle al sistema desarrollado las pruebas de caja negra o de comportamiento, el cual se centra en los requisitos funcionales del software (Pressman, 2005). Este tipo de pruebas se enfoca básicamente en el análisis de los datos de entrada y en los de salida, colocándose en el lugar del usuario, usar el sistema como él lo usaría. Esta técnica se realiza sobre la interfaz del programa a probar, además para llevar a cabo la misma no se necesita conocer la lógica interna de un programa, únicamente se verifica que la funcionalidad haga lo que debe hacer. Mediante las técnicas de prueba de caja negra se intenta encontrar errores, tales como:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para esta prueba se utilizó la técnica partición equivalente, la cual divide el dominio de entrada en clases de datos de los que se pueden derivar casos de prueba (Pressman, 2005). A continuación se muestran el caso de prueba para el RF Insertar parcela.

Caso de Prueba: Insertar Cultivo.

Descripción general: El requisito funcional inicia cuando el usuario escoge la opción de insertar un cultivo.

Condición de ejecución: Que el usuario se haya autenticado en el sistema.

Tabla 5: Caso de prueba Insertar Cultivo

Escenario	Descripción	Nombre	T. germinación	T. cosecha	T. entre fertilizante	T. entre riego	Estación	Precio	Respuesta del sistema	Flujo central
EC 1.1 Insertar Cultivo.	Permite insertar un cultivo	V	V	V	V	V	V	V	Cuando se inserta un cultivo, se muestra un mensaje de confirmación "El cultivo se ha insertado correctamente".	1- SIG para la Agricultura. 2-Seleccionar opción "Gestionar cultivo". 3-Seleccionar opción "Insertar Cultivo". 4- Se introducen los datos. 5- Seleccionar la opción "Insertar".
		Pepino	50	60	20	10	verano	5 x kg		
EC 1.2 Cancelar operación insertar cultivo.	Se cancela la operación insertar cultivo.	V	V	V	V	V	V	V	Se cancela la operación de insertar un cultivo y la información no es guardada	1- SIG para la Agricultura. 2-Seleccionar opción "Gestionar cultivo".
		lechuga	30	40	5	10	primavera	4 x lb		

									en la base de datos.	3-Seleccionar opción "Insertar Cultivo". 4- Se introducen los datos. 5- Seleccionar la opción "Cancelar".
EC 1.3	Verifica si se introdujeron todos los datos del cultivo a insertar.	-	V	V	V	V	V	-	Cuando se presiona la opción "Insertar", existe el campo nombre en blanco sale un mensaje "Debe introducir todos los datos".	1- SIG para la Agricultura. 2-Seleccionar opción "Gestionar cultivo". 3-Seleccionar opción "Insertar Cultivo". 4- Se introducen los datos. 5- Se presiona la opción "Insertar".
EC 1.5	Verifica si se introdujeron los valores correctos del	V	V	V	V	V	V	I	Cuando se presiona la opción "Insertar", no se completa la opción de insertar la parcela porque tiene campos	1- SIG para la Agricultura. 2-Seleccionar opción "Gestionar cultivo".

	cultivo a insertar.	tomate	10	35	40	10	50	Cinco \$ la lb	con valores incorrectos.	3-Seleccionar opción "Insertar cultivo". 4- Se introducen los datos. 5- Se presiona la opción "Aceptar".
--	---------------------	--------	----	----	----	----	----	----------------	--------------------------	------------------------------------------------------------------------------------------------------------------------

Tabla 6: Descripción de las variables del Caso de Prueba Insertar Cultivo

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre	Campo de texto	No	Cadena de caracteres que se corresponden al nombre del cultivo. Alfanumérico de 1 a 50 caracteres. Todo tipo de carácter.
2	T. germinación	Campo de selección	No	Numérico de 1 a 2 dígitos que representa el código identificativo del cultivo. Solo número de 0-9.
3	T. cosecha	Campo de selección	No	Numérico de 1 a 2 dígitos que representa el código identificativo del cultivo. Solo número de 0-9.
4	T. entre fertilizante	Campo de selección	No	Numérico de 1 a 2 dígitos que representa el código identificativo del cultivo. Solo número de 0-9.
5	T. entre riego	Campo de selección	No	Numérico de 1 a 2 dígitos que representa el código identificativo del cultivo. Solo número de 0-9.

6	Estación	Campo de selección	No	Campo tipo lista desplegable con los valores del cultivo.
7	Precio	Campo de texto	No	Numérico de 1 a 2 dígitos que representa el código identificativo del cultivo. Solo número de 0-9.

Para obtener mayor información puede dirigirse a los expedientes de proyecto Descripción de Casos de Prueba para cada uno de los restantes requisitos funcionales del sistema.

Resultados de las pruebas:

Las pruebas de caja negra utilizando la técnica de partición equivalente se le realizó a cada uno de los requisitos funcionales del sistema, generando un diseño de caso de prueba con sus respectivos escenarios por cada uno. En los resultados obtenidos durante la primera iteración se identificaron 24 no conformidades, la mayoría de estas relacionadas a faltas de ortografías en los mensajes mostrados al usuario y otras atadas directamente a errores en la comunicación con identidades externas al sistema, las cuales fueron corregidas. En una segunda iteración las pruebas arrojan un total de 7 no conformidades, la mayoría de estas asociadas a validaciones incorrectas en los formularios, corrigiéndose las mismas y obteniendo una tercera iteración libre de no conformidades.

A continuación en la figura 13 se muestran los resultados de las iteraciones realizadas.

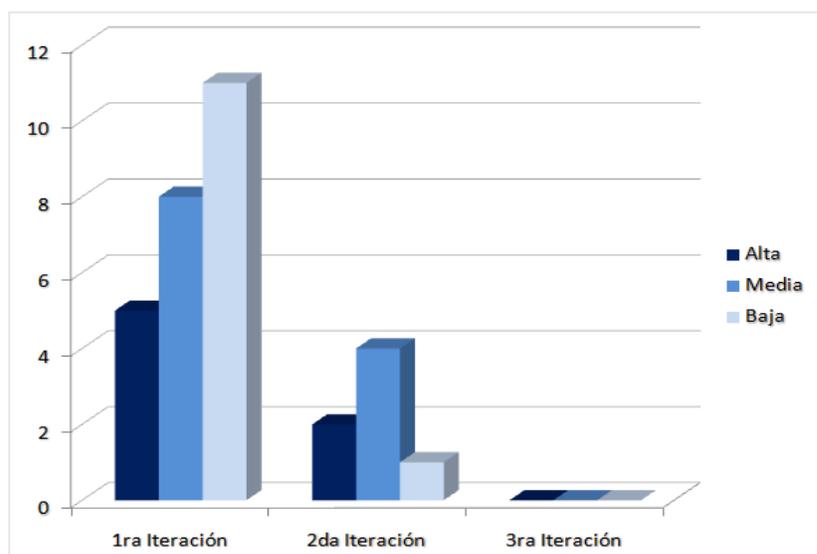


Fig 13: Prueba de caja negra

3.5 Conclusiones del capítulo

Como resultado de este capítulo se obtuvo el diagrama de componentes el cual muestra la estructura física que tiene el componente de software y el diagrama de despliegue que permitió modelar la arquitectura del sistema. Con la realización de las pruebas de caja negra aplicando la técnica de partición equivalente se determinaron 24 no conformidades en su primera iteración a las cuales se le dieron solución. Con la realización de esta prueba el Modelo de Pruebas quedó conformado por el caso de prueba Gestionar archivo multimedia de pago. Los resultados de las prueba fueron satisfactorios, lo que permitió evaluar el comportamiento del sistema y su respuesta ante las solicitudes del usuario, confirmando la validez de la solución propuesta.

Conclusiones

La realización de la investigación arroja como principal resultado, el desarrollo de un SIG para la Gestión de Cultivos, a partir de lo cual se le da cumplimiento a los objetivos propuestos:

1. Para el desarrollo de la aplicación se utilizaron las herramientas y tecnologías definidas por la plataforma GeneSIG logrando una fácil integración con la misma, así como la arquitectura y los patrones del diseño que son necesarios para garantizar el correcto funcionamiento del sistema, proporcionando uniformidad, robustez y flexibilidad a la solución desarrollada.
2. La realización de las pruebas permitió comprobar la calidad del producto y el cumplimiento de los requisitos funcionales y no funcionales establecidos.
3. El SIG desarrollado permite el análisis y procesamiento de datos espaciales georreferenciados del proceso agrícola en Cuba, proporcionando una mejor gestión y organización del mismo.

Recomendaciones

Luego de cumplidos los objetivos planteados en el presente trabajo se muestra a continuación algunas de las ideas recomendadas para trabajos futuros:

1. Crear un módulo para vincular la aplicación con sensores de humedad y GPS para utilizar datos con una mayor precisión.
2. Tomar la propuesta desarrollada como la base para un desarrollo propio de cara a clientes por el equipo de Aplicativos SIG.

Referencia Bibliográfica

Álvarez, Miguel Angel. 2001. Desarrollo web. Desarrollo web. [En línea] mayo 9, 2001. <http://www.desarrolloweb.com/php/>.

Andalucía, Junta de. 2015. Consejera de Agricultura, Ganadería y Medio Ambiente. Consejera de Agricultura, Ganadería y Medio Ambiente. [En línea] enero 29, 2015. [Citado el: octubre 20, 2015.] <http://www.juntadeandalucia.es/organismos/agriculturapescajdesarrollorural.html>.

Asenjo, Jorge Sánchez. 2012. Servidores de Aplicaciones Web. 2012.

Avila, Katty. 2014. CAVSI. Sistema Gestor de Base de Datos. CAVSI. Sistema Gestor de Base de Datos. [En línea] 2014. <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

Ayala, Raúl Gustavo Eid. 2007. EL PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN. 2007.

Booch, Grady. 2004. Object-Oriented Analysis and Design with Applications. 2004.

Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. 2000.

Braken, I. 1992. Information technology in geography and planning. Londres : s.n., 1992.

Brenes, Carlos. 2009. Sistemas de Información Geográfica. 2009.

Camacho, E, Nuñez, F y Cardeso, G. 2004. Arquitecturas de Software, Guías de Estudio. 2004.

Cowburn, Peter. 2016. php.net. php.net. [En línea] 2016. <http://php.net/manual/es/getting-started.php>.

Espinosa, A. M. y Ramos, E. P. 2012. Desarrollo de Sistemas de Información Geográfica sobre una plataforma soberana. La Habana : Universidad de las Ciencias Informáticas., 2012.

Espinosa, Antonio Membrides. 2000. Fundamentos del Mapserver, Mapscript, PostGIS y su integración con el CartoWeb. La Habana : s.n., 2000.

ESRI. 1995. ESRI . What is GIS? ESRI . What is GIS? [Online] 1995. <http://www.gis.com/content/what-gis...>

Fallas, Jorge. 2011. SISTEMAS DE INFORMACIÓN GEOGRÁFICA. 2011.

González, Dr. Armando Nova. 2009. LÍNEA DE DESARROLLO Y RESULTADOS DE LA AGRICULTURA CUBANA EN LOS ÚLTIMOS 50 AÑOS. Ciudad de la Habana : s.n., 2009.

Guerrero, Rafael Martinez. 2010. PostgreSQL. PostgreSQL. [Online] octubre 2010. http://www.postgresql.org.es/sobre_postgresql.

Hernández León, Rolando Alfredo y Coello, Sayda. 2011. El paradigma cuantitativo de la investigación científica. Ciudad de La Habana: Editorial Universitaria. ISBN 978-959-16-1307-3 : s.n., 2011.

Hernandez, Camelo. 2014. Slideshare. Slideshare. [En línea] septiembre 20, 2014. <http://es.slideshare.net/carmeloh2/metodologa-open-up-39321348>.

John, Wiley y Sons. 1997. Exploring Geographic Information Systems. 1997.

Larman, Craig. 2003. UML y Patrones. 2da Edición. 2003. 2.

MapServer. 2015. MapServer open source web mapping. MapServer open source web mapping. [Online] 2015. [Cited: 11 25, 2015.] <http://mapserver.org/es/about.html>.

Morissette, Daniel y Assefa, Yewondwossen. 2016. MapServer. MapServer. [En línea] 2016. <http://www.mapserver.org/es/mapscript/php/index-5.6.html#php5-6>.

Muñoz, Andrés Moreira. Los Sistemas de Información Geográfica y sus aplicaciones en la conservación de la diversidad biológica. VOL XII- N° 2. ISSN 0716 - 1476.

NCGIA. 1990. National Center for Geographic Information and Analysis. National Center for Geographic Information and Analysis. [Online] 1990. <http://www.ncgia.ucsb.edu/pubs/pubslst.html#98-1>.

Olaya, Victor. 2010. Sistemas de Información Geográfica. Girona : s.n., 2010.

Ortiz, Gabriel. 2002. ¿Qué son los Sistemas de Información Geográfica? . ¿Qué son los Sistemas de Información Geográfica? . [En línea] 2002. [Citado el: febrero 15, 2016.] <http://www.gabrielortiz.com/art.asp?Info=012#Topolog%C3%ADAs>.

Paradigm, Visual. 2008. Visual Paradigm. Visual Paradigm. [Online] 2008. <http://www.visual-paradigm.com/product/vpumf/features/>.

PostGIS. 2015. PostGIS. PostGIS. [Online] 2015. <http://postgis.net/>.

Pressman, Roger S. 2005. Ingeniería de Software. Un enfoque práctico. 2005.

Pressman, Roger. 2011. Software Engineering, a practitioner's approach. (7ª edición). 2011. ISBN 9780071267823.

Programación, Lenguajes de. 2009. Lenguajes de programación. Lenguajes de programación. [En línea] 2009. <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.

Ravelo Ortega, MSc. Reinaldo y León Noguera, MSc. Pedro. 2007. Fitotecnia General. Aplicada a las Condiciones Tropicales. La Habana : s.n., 2007.

Reynoso, Carlos y Kiccillof, Nicolás. 2004. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. . Buenos Aires : s.n., 2004.

Robaina, I. 2008. Propuesta del Diseño Arquitectónico del Simulador de Sistemas Biológicos. 2008.

Rodríguez, D y Fernández, R García. 2009. Plataforma para Formación de Investigadores a Distancia. 2009.

Rojas Martínez, Omara, et al. 2014. Implementation of a geographic information system for surgacane production in Santiago de Cuba. Santiago de Cuba : s.n., 2014.

Rosas, Juan Eladio Sánchez. 2008. Desarrollo en Web. Desarrollo en Web. [En línea] 10 22, 2008. [Citado el: 11 25, 2015.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs>.

Sánchez, Tamara Rodríguez. 2015. Metodología de desarrollo para la actividad productiva de la UCI. La Habana : s.n., 2015.

Sommerville, Ian. 2005. Ingeniería de Software, séptima edición. Madrid : s.n., 2005.

Star, J. and Estes, J. 1990. Geographical Information Systems: An Introduction. 1990.

Tejerina, Isabel García. 2014. Ministerio de la Agricultura, Alimentación y Medio Ambiente. Ministerio de la Agricultura, Alimentación y Medio Ambiente. [En línea] 01 29, 2014. [Citado el: 01 19, 2016.] <http://www.magrama.gob.es/es/agricultura/temas/sistema-de-informacion-geografica-de-datos-agrarios/>.

Titular, Dra. C. María Elena Guardo García. Prof. and Auxiliar, Dr. C. Edmundo Claudio Pérez . Prof. 2013. THE SCIENTIFIC PROBLEM: ITEMS THAT IDENTIFY AND CHARACTERIZED AS A COMPONENT OF THE THEORETICAL DESIGN OF A SCIENTIFIC INVESTIGATION. 2013.

UML, Tutorial. 2007. sparxsystems. sparxsystems. [Online] 2007. [Cited: mayo 10, 2015.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Valdés, Damián Pérez. 2007. Maestros del web. Maestros del web. [En línea] julio 3, 2007. <http://www.maestrosdelweb.com>.

Viera, Deborah González, et al. 2012. Technological management for the rice crop with an agroecological and participative approach in a local scale. PART I.- Diagnosis, strategy design and proposal of action plan. Ciudad de la Habana, Bauta, Artemisa : s.n., 2012. ISSN papel: 0253-5785.

Zayas, Dr. Cs. Carlos Álvarez de. 1995. Metodología de la investigación científica. Santiago de Cuba : s.n., 1995.

Anexos

Anexo 1: Las entrevistas se realizaron con el objetivo de investigar toda la información referente a la agricultura para conocer cómo se gestiona la misma en nuestro país.

Estas entrevistas se les realizaron a varios trabajadores y especialistas del MINAG, así como a campesinos independientes.

Preguntas:

1. ¿Cómo se realiza el manejo de la información referente al proceso agrícola en Cuba?
2. ¿Cuáles son los requisitos fundamentales que debería cumplir la aplicación a desarrollar?
3. Aspectos a tener en cuenta para analizar y procesar una parcela determinada.
4. ¿Cómo se clasifican los tipos de suelo?
5. Para llevar un correcto control de los cultivos que es importante conocer de los mismos.
6. ¿Qué aspectos dentro del proceso agrícola le resulta más complejo?

Respuestas:

1. El análisis y procesamiento del proceso agrícola en Cuba se realiza de forma semi-automatizada, aunque existen algunos software como el SIG para las decisiones en las producciones cañeras de Santiago de Cuba todavía quedan algunos procesos sin automatizar, como la localización geográfica de otros tipos de cultivos. Además, algunos de los aspectos más importantes como el control de plagas, riego y fumigación se realizan de forma manual, provocando en ocasiones lamentables pérdidas de información.
- Algunos requisitos funcionales con los que debe cumplir la aplicación son los siguientes: gestionar cultivo, parcela, transporte, insecticida, fertilizante, alerta y plaga.

2. Para el análisis y procesamiento de una parcela es necesario conocer su área, el tipo de suelo, la humedad, y además el tiempo que lleva sin sembrar o si no ha sido sembrada anteriormente para definirla como ociosa o no.
3. Los suelos se clasifican en limosas, turbas, salinos y arcillosas.
4. Es necesario conocer de los cultivos para tener un correcto control de los mismo en que estación del año se debe sembrar, el tiempo de germinación, el tiempo de cosecha, el tiempo entre fertilizantes y el tiempo entre riego.
5. Dentro del proceso agrícola se dificultan en gran medida la gestión del transporte para la recogida de los productos. Definir que fertilizantes e insecticidas aplicar cuando tenemos una gama amplia de cultivos.

Anexo 2: A continuación se presentan imágenes de la aplicación desarrollada, SIG para la Gestión de Cultivos en Cuba.

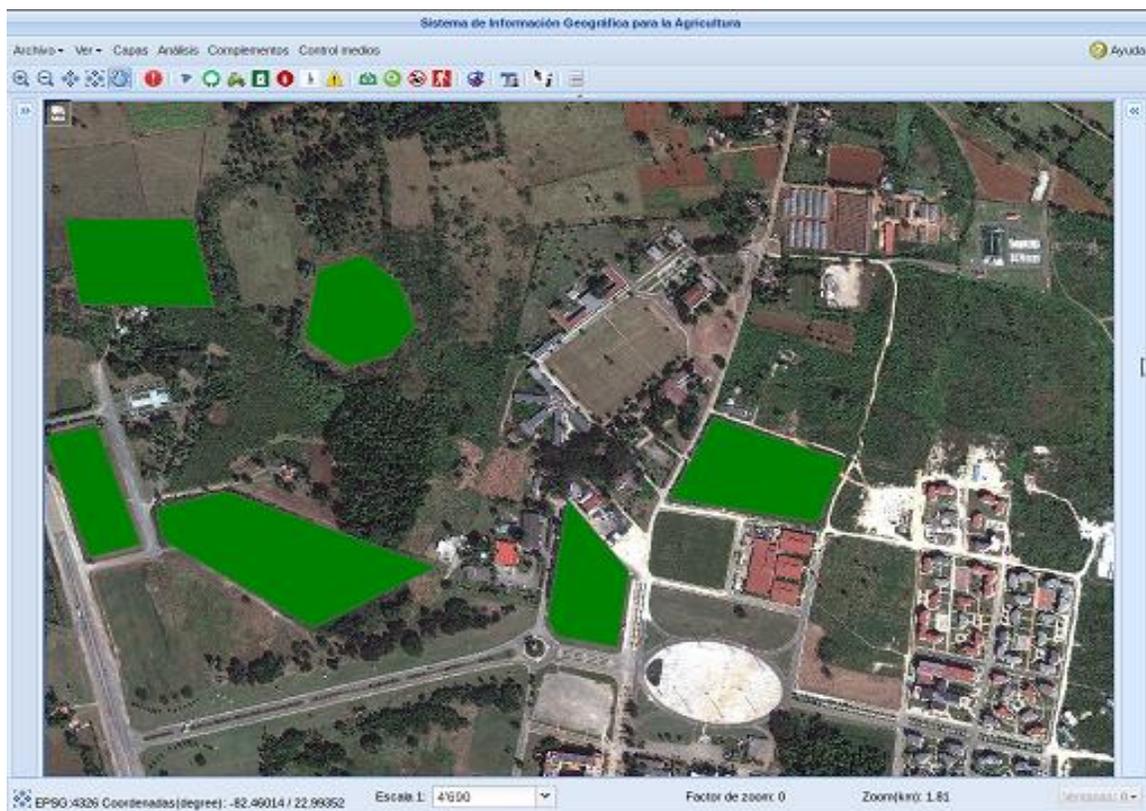


Fig 14: Interfaz principal del SIG para la Gestión de Cultivos

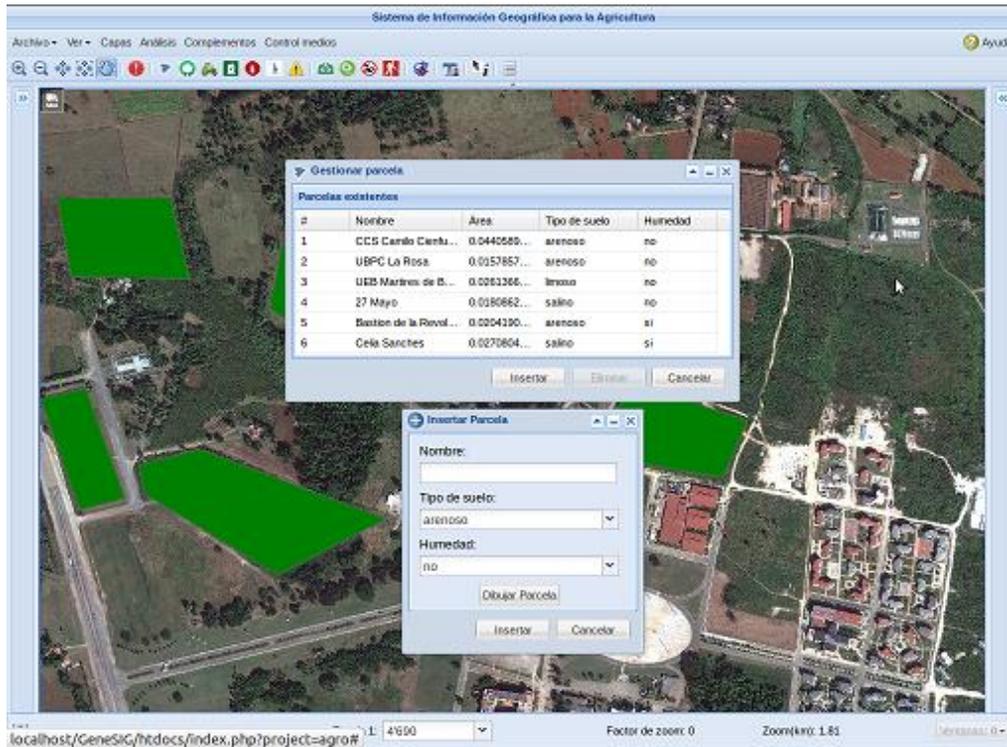


Fig 15: Interfaz del SIG para Gestionar parcela

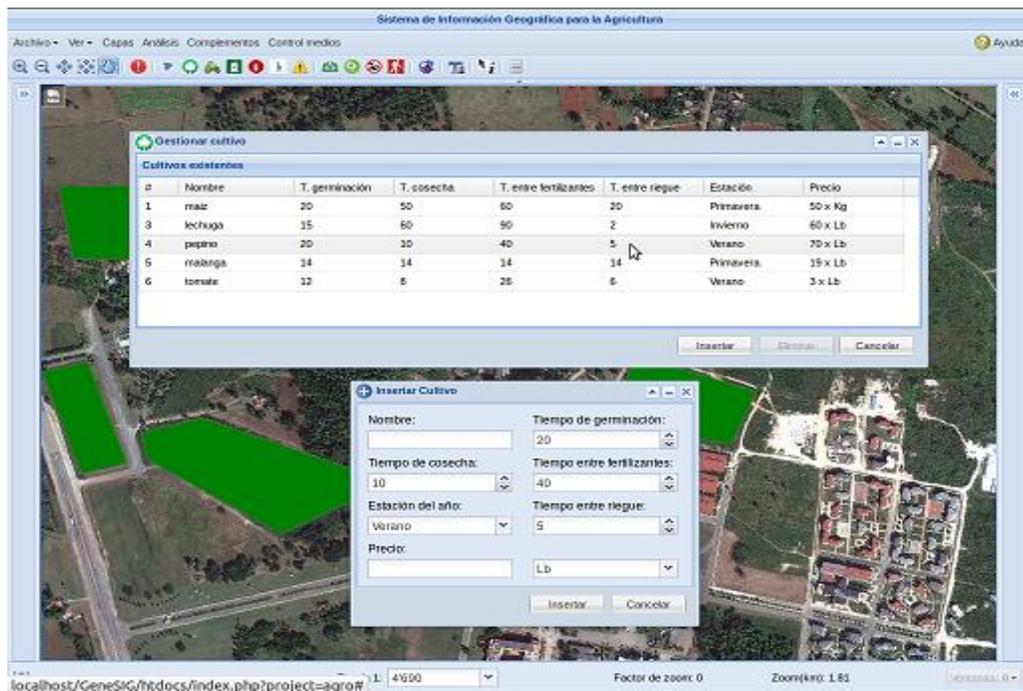


Fig 16: Interfaz del SIG para Gestionar cultivo

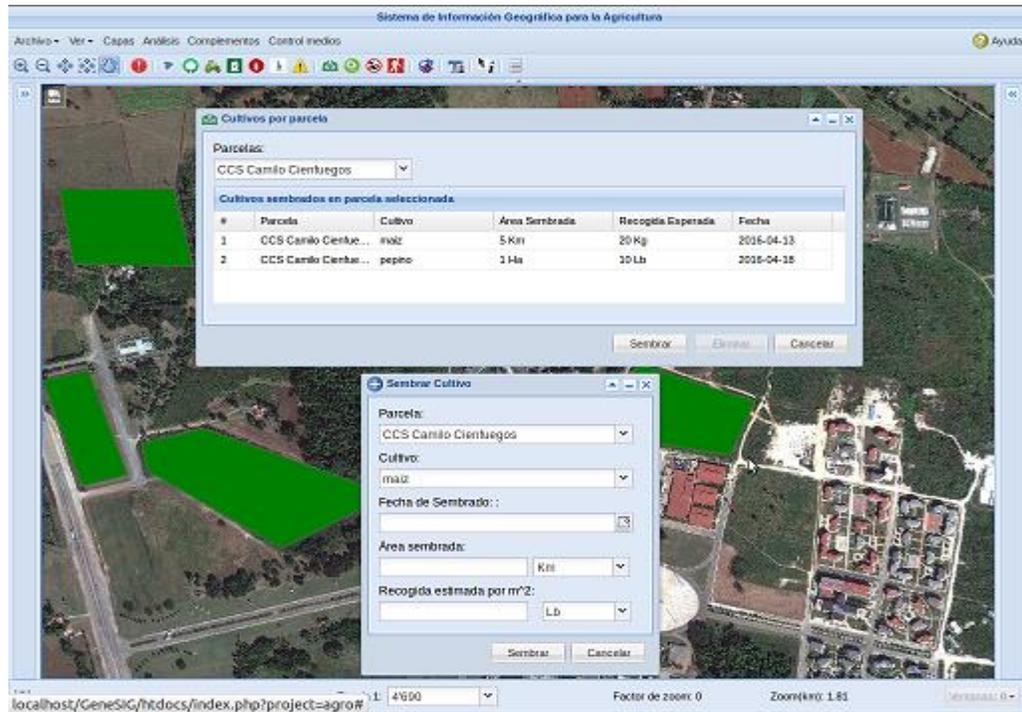


Fig 17: Fig 17: Interfaz del SIG para sembrar un cultivo en una parcela determinada

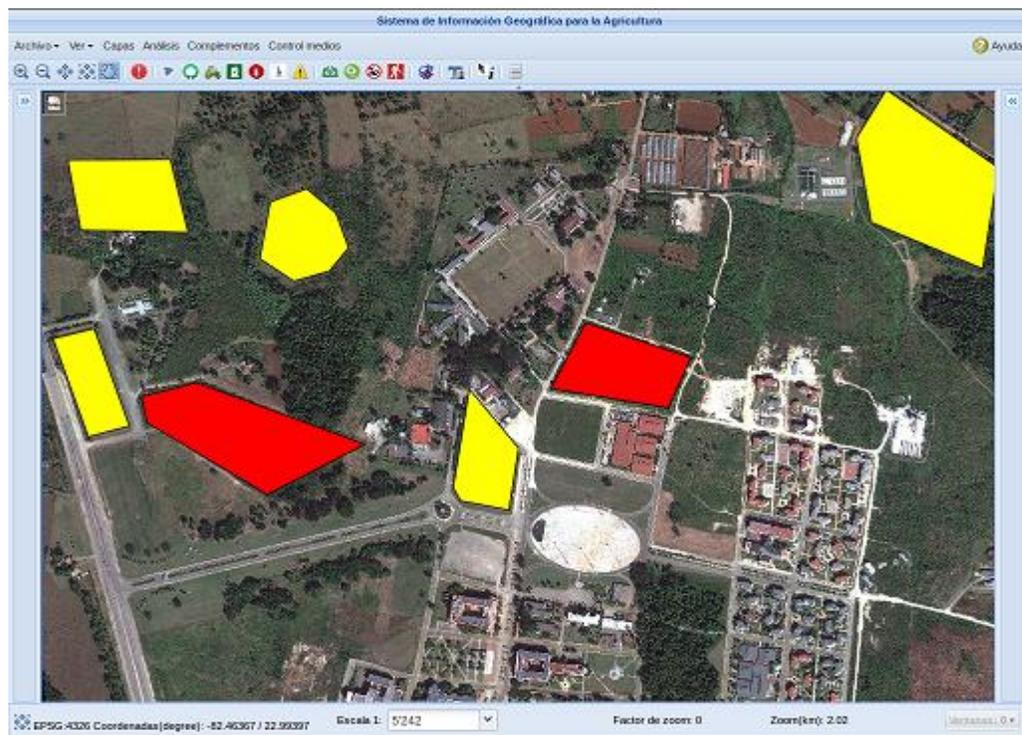


Fig 18: Interfaz del SIG para mostrar los cultivos afectados por una plaga determinada