

Universidad de las Ciencias Informáticas
FACULTAD 6



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

Título: *Módulo de Georeferenciación Eventual del Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas.*

Autor(es): Julio César Rodríguez Torres.

Tutor(es): Ing. Pedro Enrique Palau Isaac.

La Habana Junio, 2013



El destino es la cuartada sigilosa del que lo pretende todo y nunca acierta

Ricardo Arjona

Declaración de Autoría

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Julio Cesar Rodríguez Torres

Ing. Pedro Enrique Palau Isaac

Firma del Autor

Firma del Tutor



Datos de Contacto

Datos de Contacto

TUTOR: Ing. Pedro Enrique Palau Isaac.

Categoría docente: Instructor.

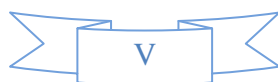
Universidad de las Ciencias Informáticas, Habana, Cuba.

e-mail: pepalau@uci.cu

Agradecimientos

Agradecimientos

Agradezco en primer lugar a mi familia por ser las personas que son y por el apoyo incondicional brindado durante toda mi vida. A todos mis amigos, tanto los que hoy se encuentran aquí como a los que no están por una u otra razón.



Dedicatoria

Dedicatoria

Dedicada a mi familia puesto que sin su apoyo no hubiera logrado llegar hasta aquí, a los amigos verdaderos que siempre han estado dándome la mano y animándome en los momentos en que estuve casi quebrado, en fin a todas las personas que confiaron en mí aun en los momentos más difíciles cuando todo se tornó gris.

Resumen

Resumen

Con el desarrollo de software y herramientas que ayudan a mejorar la toma de decisiones y el almacenamiento de información, viene aparejado el concepto de Sistemas de Información Geográfica (SIG) que permiten representar información geográficamente referenciada. La Universidad de las Ciencias Informáticas se convierte en un patrón a seguir en cuanto al desarrollo de SIG se refiere. Cuenta con el SIG-UCI que permite la búsqueda de personas, lugares, el cálculo de rutas y otras funcionalidades pero ninguna de ellas encaminadas a la representación de la organización de los eventos que tienen lugar dentro de ella.

Con el presente trabajo se propone el desarrollo de una aplicación que servirá como solución a esta carencia y se representarán los principales artefactos generados durante la investigación, partiendo de una problemática y basándonos en el estado del arte de soluciones existentes. Bajo la concepción del patrón arquitectónico Modelo Vista Controlador, la metodología de desarrollo Proceso Unificado de Desarrollo de Software y herramientas de código abierto.

PALABRAS CLAVE

Eventos, herramientas, referenciada, SIG.

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1: ELEMENTOS TEÓRICOS DE LA INVESTIGACIÓN.....	5
1.1. Introducción	5
1.2. Conceptos Asociados.....	5
1.3. Reseña histórica sobre el surgimiento de los SIG	6
1.4 Situación problemática	7
1.5 Análisis de soluciones existentes	8
1.6 Conclusiones Parciales	11
CAPÍTULO 2: SELECCIÓN Y DESCRIPCIÓN DE HERRAMIENTAS Y METODOLOGÍAS.....	12
2.1. Introducción	12
2.2. Metodologías de Desarrollo de Software.....	12
2.3. Lenguaje de Modelado UML v2.0.....	13
2.4. Herramienta CASE Visual Paradigm v8.0	13
2.5. Lenguaje de programación PHP5.....	13
2.6. Sistema de Gestión de Bases de Datos (PostgreSQL v8.4).....	14
2.6.1. PostGIS v1.5.3	14
2.7. Servidor de Aplicaciones Web Apache2.....	14
2.8. Servidor de Mapas MapServer v5.2.2	15
2.9. Lenguaje de programación del lado del cliente (JavaScript).....	15

Índice

2.9.1 Librería de JavaScript (Ext. JS v4.0)	15
2.9.2 Librería de JavaScript (OpenLayers v2.10)	16
2.10. Framework de desarrollo (Symfony v1.4)	16
2.11. Entorno de Desarrollo (NetBeans v7.1)	17
2.12. Conclusiones.....	17
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	18
3.1. Introducción	18
3.2. Modelo del Dominio	18
3.2.1. Diagrama de clases del Modelo del Dominio	19
3.2.2. Conceptos tratados en el modelo del dominio	19
3.3. Requisitos	20
3.3.1. Requisitos funcionales.....	20
3.3.2 Requisitos no funcionales.....	22
3.4. Casos de uso del sistema	24
3.4.1 Diagrama de casos de uso del sistema	25
3.5. Conclusiones	28
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	30
4.1. Introducción.	30
4.2 Descripción de la Arquitectura.....	30
4.2.1. Ventajas que presenta este estilo.....	31
4.2.2 Patrones de diseño.....	31

Índice

4.2.3. Diagrama de clases del diseño.....	35
4.3. Diseño de la base de datos.....	37
4.5 Modelo de implementación.....	40
CONCLUSIONES.....	46
RECOMENDACIONES.....	47
REFERENCIAS BIBLIOGRÁFICAS.....	48
ANEXOS.....	54
GLOSARIO DE TÉRMINOS.....	68

Índice de Figuras

Imagen 1: Ejemplo del GvSIG	9
Imagen 2: Genesig.....	10
Imagen 3: SIG-UCI	10
Imagen 4: Modelo de clases del dominio	19
Imagen 5: Diagrama de Casos de Uso	25
Imagen 6: Diagrama de clases del diseño gestionar áreas	36
Imagen 7: Diagrama Entidad-Relación.....	38
Imagen 8: Diagrama de despliegue del Módulo de Georeferenciación Eventual	39
Imagen 9: Diagrama de componentes del Módulo de Georeferenciación Eventual.....	41
Imagen 10 Diagrama de clases del diseño del CU Administrar Imágenes	65
Imagen 11 Diagrama de clases del diseño del CU Gestionar eventos	66
Imagen 12 Diagrama de clases del diseño del CU Mostrar Información	67

Índice de Tablas

Tabla 1: Requisitos funcionales	21
Tabla 2 Descripción del CU Gestionar áreas	28
Tabla 3: Secciones a probar en el CU Gestionar evento.....	43
Tabla 4: Descripción de variables	43
Tabla 5: Matriz de datos del CU Gestionar eventos	44
Tabla 6 Descripción del CU Gestionar Evento	57
Tabla 7 Descripción del CU Administrar imágenes	59
Tabla 8 Descripción del CU Mostrar información	61
Tabla 9 Descripción del CP Gestionar áreas.....	62
Tabla 10 Descripción del CP Administrar imágenes.....	63
Tabla 11 Descripción del CP Mostrar información.....	64

Introducción

INTRODUCCIÓN

Debido al creciente auge de las Tecnologías de la Información y las Comunicaciones (TIC) el mundo moderno se ha ido encaminando hacia una revolución tecnológica que asombra cada vez más a todos los habitantes del planeta. Las TIC tratan sobre el empleo de computadoras y aplicaciones informáticas para transformar, almacenar, gestionar, proteger, difundir y localizar los datos necesarios para cualquier actividad humana.

En Cuba existen centros tanto laborales como estudiantiles que aplican el trabajo con las TIC, donde la Universidad de las Ciencias Informáticas (UCI) juega un papel sumamente importante, puesto que cuenta con personal calificado para el estudio y posterior desarrollo de las TIC. La UCI cuenta con facultades que se especializan en el empleo de diferentes herramientas para lograr el desarrollo de aplicaciones que serán utilizadas para automatizar procesos en distintas ramas de la economía del país, con el fin de hacer más eficiente el trabajo diario de un sin número de personas. En el Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6, se encuentra el Departamento de Geoinformática que se especializa en el trabajo con los Sistemas de Información Geográfica (SIG).

Un SIG se define como *“un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos y dar solución a los problemas complejos del manejo y planeamiento territorial”*(10). El SIG permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla. Estos sistemas informáticos poseen un carácter versátil y pueden resolver cuestiones tales como la localización, condición, rutas y modelo.

Ante la creciente necesidad de informatizar procesos para hacer más ameno y factible el trabajo de los residentes y trabajadores, la UCI cuenta con un SIG (SIG-UCI) que ofrece funcionalidades que facilitan la geolocalización de estructuras (edificios, docentes, comedores) y la obtención de información sobre cada una de las entidades u objetos que se encuentran en la misma. Debido a que la universidad se encuentra inmersa en cada momento, en la realización de eventos o actividades políticas, científicas, deportivas y culturales que se desarrollan dentro de la entidad, que no se cuenta con una herramienta que sea capaz de representar, gestionar y almacenar información referente a los bloques de estudiantes participantes en

Introducción

estas actividades ni a su ubicación territorial puesto que esto se realiza de forma manual utilizando documentos en formato Word, Excel o PDF. Además la distribución de la información de la planificación se realiza haciendo uso del correo electrónico lo cual potencia el tráfico de paquetes innecesarios en la red que puede llevar a la congestión de la misma y por tanto a que la información llegue tarde a su destino. Se crea el Módulo de Georeferenciación Eventual el cual será desarrollado sobre la plataforma Atlas e integrado al SIG-UCI y que tiene como objetivo permitir representar y almacenar información relevante referida a los eventos para así disponer de ella en cualquier momento y visualizar una representación gráfica de los mismos. Cada evento representado será almacenado en forma de una capa temática que podrá ser activada por los usuarios en el momento en que deseen tener información sobre un determinado evento.

Luego de analizados estos elementos, se define como **problema a resolver** la interrogante de: ¿cómo facilitar la visualización y divulgación de la información referente a los distintos eventos que se desarrollan en la Universidad de las Ciencias Informáticas?

Para dar solución a este problema se determinó como **objeto de estudio** la georeferenciación en los Sistemas de Información Geográfica. Se define como **campo de acción** la georeferenciación eventual en el SIG-UCI.

La meta que se pretende alcanzar en la presente investigación para dar solución al problema planteado, se concreta según el **objetivo general**: Desarrollar un módulo para la representación geográfica eventual para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas. Definiéndose como **idea a defender** que el desarrollo de un módulo de georeferenciación eventual para el SIG-UCI, permitirá la representación gráfica de la planificación de los eventos, así como el acceso a sus principales características.

Para darle cumplimiento al objetivo general se han propuesto las siguientes **tareas de investigación**:

- Fundamento de tendencias actuales, tecnologías y conceptos más importantes relacionados con el proceso de georeferenciación en los Sistemas de Información Geográfica.
- Análisis de SIG que tengan funcionalidades relacionadas con la georeferenciación.

Introducción

- Selección de las tecnologías y herramientas a utilizar en el desarrollo del módulo.
- Identificación de los principales conceptos asociados al dominio del problema.
- Obtención de los artefactos generados por la metodología de software escogida.
- Modelación del diseño de la aplicación informática utilizando para ello patrones de diseño.
- Implementación del módulo.
- Validación del módulo implementado mediante las pruebas de Caja Negra.

Durante el desarrollo de la investigación, se emplearon diferentes métodos de investigación para la obtención de la información necesaria. Los métodos teóricos usados en la investigación se describen a continuación:

Histórico-Lógico: Empleado durante la primera fase de la investigación, ya que primeramente se analizó el desarrollo de las tecnologías y los principales conceptos asociados a las mismas. Se hizo énfasis además en el surgimiento de los Sistemas de Información Geográfica y el desarrollo y uso de estos en el país.

Analítico-Sintético: Utilizado en el momento del análisis de la información referente a las herramientas utilizadas para el desarrollo del módulo. Permitió la realización de una síntesis de las principales características de estas tecnologías.

Modelación: Después de analizada la información y el desarrollo de la investigación, se procedió a realizar los modelos correspondientes al ciclo de vida del Módulo de Georeferenciación Eventual. Este método facilitó la realización de las tareas de diseño de los procesos y la implementación del sistema.

Método de investigación empírico

Entrevista, puesto en práctica en las diferentes entrevistas realizadas al líder del proyecto y al personal calificado, con el objetivo de obtener información básica sobre el ambiente en que se desarrolló el trabajo investigativo. Para ello se realizó una entrevista al arquitecto de la línea de desarrollo de software y líder

Introducción

del equipo de integración y componentes del proyecto Aplicativos SIG Alain León Companioni, la cual arrojo como resultado todos los requisitos funcionales.

El trabajo de diploma se estructura en cuatro capítulos que se describen a continuación:

Capítulo 1: Elementos teóricos de la investigación.

Contempla la fundamentación teórica de todo el trabajo investigativo, donde se exponen conceptos que permitirán familiarizarse con términos utilizados en el tema de investigación, tales como la cartografía, georeferenciación y sistemas de información. Además se especifica de forma más detallada la situación problemática y el problema científico a resolver, se mencionan y analizan otras soluciones existentes.

Capítulo 2: Selección y descripción de herramientas y tecnologías

En este capítulo se muestran las características fundamentales de las principales tecnologías, herramientas y lenguajes que se utilizan en el mundo para la implementación de este tipo de sistemas. Se realizará también la selección de las que se utilizarán para la construcción de la solución propuesta.

Capítulo 3: Análisis y Diseño del sistema

Los requisitos funcionales y no funcionales del sistema, los casos de uso derivados de ellos y su descripción serán los temas abordados en este capítulo. Además se representa el modelo de dominio y el diagrama de caso de uso correspondiente al sistema.

Capítulo 4: Implementación y Prueba

Contempla los aspectos fundamentales sobre la construcción del sistema propuesta en el capítulo anterior, reflejando los principales componentes y diagramas usados, además de la descripción de las pruebas realizadas al mismo.

Capítulo 1: Elementos Teóricos de la Investigación

CAPÍTULO 1: ELEMENTOS TEÓRICOS DE LA INVESTIGACIÓN.

1.1. Introducción

Este capítulo comprende toda la teoría relacionada con los Sistemas de Información Geográfica, además se hace referencia a los diferentes conceptos de términos que comprende la investigación. Se describen el objeto de estudio del trabajo y el entorno en que se desarrolla el proceso. Se hace un análisis de las principales soluciones nacionales e internacionales existentes, cuyas funcionalidades resultan similares a las que se desean automatizar con la solución propuesta y se aborda también, de forma más detallada, la situación problemática que da origen a este trabajo.

1.2. Conceptos Asociados

Cartografía:

“La cartografía es la ciencia que se encarga del trazado y el estudio de mapas geográficos, por lo cual se puede decir que esta ciencia se ocupa de la organización y representación de la información espacial para la creación de mapas referentes a la información contenida en la superficie terrestre”. (5)

“Es la disciplina que se ocupa de la organización, representación, comunicación y utilización de la información espacial con vistas a la elaboración de mapas y otros tipos de representación convencional gráfica de la Tierra o cualquier parte del Universo. La ciencia que tiene por objeto el establecimiento de las normas, técnicas y procedimientos que estudian diversas realidades o fenómenos (urbanos, culturales, económicos, étnicos, políticos, entre otros.) y la confección de planos y mapas”. (23)

Se puede decir que la cartografía no es más que la ciencia que se encarga de la elaboración de mapas geográficos.

Georeferenciación:

“Es el posicionamiento en el que se define la localización de un objeto espacial en un sistema de coordenadas. Este proceso es utilizado en los Sistemas de Información Geográfica ya que constituye la función principal de estos.” (5)

Capítulo 1: Elementos Teóricos de la Investigación

“Proceso mediante el cual se logra una definición geográfica precisa de la ubicación de puntos, líneas y polígonos presentes en un mapa o foto, gracias a la correlación de estos y sus respectivos representados en un sistema de coordenadas reales” (24).

Se puede decir que la georeferenciación es el proceso mediante el cual se hace referencia a las coordenadas de un punto ubicado en un mapa que se encuentra en cualquier lugar de la superficie terrestre.

Sistema de Información:

“Conjunto de elementos que interactúan entre sí para procesar y distribuir la información” (5).

1.3. Reseña histórica sobre el surgimiento de los SIG

En su surgimiento, el hombre pintaba en las paredes de las cuevas, los animales que cazaban asociados con líneas que se piensa correspondían con las rutas de migración de las especies. Estos primeros antecedentes simulan la característica de los SIG modernos de tener una imagen asociada con un atributo de información. En 1854 el Dr. John Snow cartografió la incidencia de los casos de cólera en un mapa de Soho, un distrito de Londres. Este “protoSIG” constituye el ejemplo más antiguo del método geográfico. El año 1962 vio la primera utilización real de los SIG en el mundo, concretamente en Ottawa (Ontario, Canadá), a cargo del Departamento Federal de Silvicultura y Desarrollo Rural. Desarrollado por Roger Tomlinson, el llamado SIG de Canadá, fue una iniciativa orientada a la gestión de los vastos recursos naturales del país con información cartográfica relativa a tipos y usos del suelo, agricultura, espacios de recreo, vida silvestre, aves acuáticas y silvicultura.

Durante los últimos 20 años los sistemas de información geográfica han constituido una de las herramientas de trabajo más utilizadas por los investigadores, analistas, planificadores y otros. Los SIG tienen como objetivo el manejo de información relacionada con diversos niveles de información espacial o territorial. Entre las múltiples definiciones de SIG se puede encontrar que: *“Es un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados (georeferenciados), para la solución de los problemas complejos del manejo y planeamiento territorial” (2).*

Capítulo 1: Elementos Teóricos de la Investigación

La tecnología de los Sistemas de Información Geográfica puede ser utilizada para investigaciones científicas, la gestión de los recursos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, entre otros. La composición de los SIG está dividida en cinco partes que son fundamentales, el hardware, el software, la Información, el Personal capacitado y los métodos de desarrollo.

Sus componentes principales son:

- Sistema de manejo de base de datos.
- Interfaz Gráfica de Usuarios.
- Herramientas para captura y manejo de información geográfica.
- Herramientas para soporte de consultas, análisis y visualización de datos geográficos

En un SIG la información puede ser almacenada en formato ráster o vectorial:

- El modelo de SIG ráster se centra en las propiedades del espacio más que en la precisión de la localización. Divide el espacio en celdas regulares donde cada una de ellas representa un único valor. Cuanto mayor sean las dimensiones de las celdas (resolución) menor es la precisión o detalle en la representación del espacio geográfico (5).
- En el caso del modelo de SIG vectorial, el interés de las representaciones se centra en la precisión de localización de los elementos sobre el espacio y donde los fenómenos a representar son discretos, es decir, de límites definidos (5).

1.4 Situación problemática

La Universidad de las Ciencias Informáticas representa uno de los pilares fundamentales dentro de los cuales se sostiene la industria del software en el país, esto lleva consigo que sea punto de referencia a la hora de hablar de procesos informatizados. Basados en el creciente avance tecnológico, en la importancia de los SIG y lo que representan estos para la toma de decisiones es que surge la necesidad de informatizar los procesos independientemente de que puedan resultar sencillos desde algún punto de vista. Dentro de la UCI tienen lugar una serie de eventos (marcha de las antorchas, concentraciones para

Capítulo 1: Elementos Teóricos de la Investigación

la salida del centro, juegos deportivos, etc.), y se requiere que sea representada la organización de la masividad de estudiantes que participan en los mismos.

La Universidad cuenta con el SIG-UCI el cual muestra un mapa detallado de todas sus áreas, brinda una serie de funcionalidades tales como el cálculo de distancias entre puntos y brinda además información referente al personal que convive en la universidad. Con el fin de contar con una herramienta que permita representar la organización de las personas participantes en los diferentes eventos y garantizarle al encargado de velar por la planificación y correcto funcionamiento de los mismos una representación visual de cómo estarán distribuidas. Además de evitar el tráfico innecesario de paquetes en la red puesto que este proceso se realiza mediante la vía correo electrónico, surge el Módulo de Georeferenciación Eventual que se desarrollara sobre la plataforma Atlas y que será integrado al software SIG-UCI.

1.5 Análisis de soluciones existentes

GvSIG Desktop:

Es un SIG desarrollado por la comunidad Valenciana bajo la licencia GPL, distribuido de forma libre, es multilinguaje y multiplataforma. Este SIG permite el trabajo con algunas opciones básicas sobre los mapas (**acerca y alejar el mapa**), además de que cuenta con funcionalidades tales como el acceso a base de datos, tablas de tipo PostGIS y la edición gráfica la cual permite: añadir capa de eventos, insertar punto, polígonos, líneas entre otros y personalizarlos. Este software está basado en tecnología de software libre y presenta otras funcionalidades encaminadas a otro tipo de trabajo que no resultan de mayor ayuda para la solución a desarrollar. La principal característica por la cual no puede ser utilizado la constituye el propio hecho de ser una aplicación desktop, además de que para su desarrollo se utilizaron herramientas y tecnologías que no permitirán una correcta integración al SIG-UCI.

Capítulo 1: Elementos Teóricos de la Investigación



Imagen 1: Ejemplo del GvSIG



Plataforma Génesis:

Está implementada sobre herramientas y tecnologías libres, lo que facilita su uso y manejo de acuerdo a las necesidades específicas de sus usuarios. Presenta una interfaz bastante sencilla y de fácil manejo. Cuenta también con el Módulo Edit el cual permite la edición de objetos puntuales, lineales y poligonales sobre el mapa, con atributos y simbología personalizada. Esta plataforma está desarrollada sobre el framework CartoWeb y sus módulos no se pueden integrar al SIG-UCI puesto que este ha sido desarrollado sobre la plataforma Atlas.

Capítulo 1: Elementos Teóricos de la Investigación



Imagen 2: Plataforma GeneSIG.

SIG-UCI:




De este SIG fueron analizados los módulos de localización y cálculo de rutas. El estudio de la implementación de dichos módulos resulta de suma importancia para la implementación, puesto que sirvieran de base para el desarrollo del Módulo de Georeferenciación Eventual.



Imagen 3: SIG-UCI

Capítulo 1: Elementos Teóricos de la Investigación

1.6 Conclusiones Parciales

Los conceptos asociados al dominio que fueron expuestos y la breve reseña histórica abordando el surgimiento de los SIG ofrecen herramientas teóricas importantes tanto culturales como técnicas para la comprensión del  nio del problema. A pesar de que las soluciones similares existentes no dan respuestas por sí solas a la problemática planteada ni se pueden adecuar íntegramente a las características particulares del  d de Georeferenciación Eventual, su análisis aporta elementos teóricos y una vista aceptada  s a tener en cuenta a la hora del desarrollo del módulo. Estas soluciones no dan respuestas por sí solas a la problemática planteada ni se pueden adecuar íntegramente a las características particulares del Módulo de Georeferenciación Eventual.

Capítulo 2: Selección y Descripción de Herramientas y Tecnologías

CAPÍTULO 2: SELECCIÓN Y DESCRIPCIÓN DE HERRAMIENTAS Y METODOLOGÍAS

2.1. Introducción



En la actualidad el desarrollo de la tecnología, ofrecen una amplia gama de posibilidades para que grandes y medianas empresas puedan darle soluciones a sus necesidades. En el presente capítulo se fundamentan las tecnologías utilizadas para la construcción del sistema que dará solución a la problemática que dio origen a la investigación. Este grupo de tecnologías lo conforman: la metodología para guiar el proceso de desarrollo del sistema, el lenguaje de modelado que permitirá el entendimiento del proceso de desarrollo mediante el uso de diagramas, las herramientas CASE que soporten dicho modelado, el gestor de base de datos escogido, el lenguaje de programación, el entorno de desarrollo y el framework que soportará la aplicación.

La selección de estas herramientas y la metodología se debió principalmente al hecho de que se estará desarrollando un módulo que posteriormente se integrará al Sistema de Información Geográfica de la UCI, el cual ha sido desarrollado sobre la plataforma Atlas utilizando estas herramientas y se mantendrá su estructura arquitectónica.

2.2. Metodologías de Desarrollo de Software

El proceso de desarrollo del software debe estar estructurado de forma correcta y disciplinada para de esta forma evitar un derroche de tiempo y recursos. Es preciso definir una estrategia para darle un orden a las posibles tareas a desarrollar. Existen disímiles metodologías de desarrollo, las cuales se dividen en dos grandes grupos y hacen referencia a los procesos de desarrollo de software: las metodologías robustas o tradicionales y las metodologías ágiles.

La metodología robusta RUP se define como metodología de desarrollo a utilizar puesto que esta, tiene como objetivo el lograr un desarrollo eficiente, está dirigido por casos de usos y centrado en la arquitectura, iterativo e incremental. Además de que esta genera una amplia documentación de todas las fases de desarrollo del producto lo cual facilita las futuras actualizaciones (5). RUP es una metodología pesada, pensada para grandes equipos de desarrollo y debido a la gran cantidad de documentación que genera el trabajo con ella se torna más lenta a pesar de esto se utiliza ya que los artefactos que se generan son de gran importancia para la universidad puesto que los equipos de desarrollo varían con

Capítulo 2: Selección y Descripción de Herramientas y Tecnologías

facilidad y estos serán fundamentales para futuras actualizaciones, además para ser consecuente con la metodología utilizada en el SIG-UCI.

2.3. Lenguaje de Modelado UML v2.0.

El lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de software más usado en la actualidad, este permite visualizar, especificar, construir y documentar un sistema a través de diagramas. Este a su vez también incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. El UML permitirá que tanto los programadores, analistas o cualquier otra persona interesada, sean capaces de entender el funcionamiento del sistema a partir de los esquemas que serán generados (5).

2.4. Herramienta CASE Visual Paradigm v8.0

Durante la realización del proyecto la herramienta de modelado que se utilizó fue el Visual Paradigm. El uso de esta herramienta permitió crear los diferentes tipos de diagramas que fueron utilizados para dar la representación visual: los procesos del negocio, la base de datos y las clases del sistema de manera visual. Soporta patrones de diseño para lograr mejores prácticas además permite la configuración de los estilos y formatos de los diagramas incorporando imágenes y estereotipos, así como la exportación de los diagramas en formato de imagen. Esta herramienta soportó el ciclo de vida completo del desarrollo del software: modelado, implementación, pruebas y despliegue. No se utilizó Rational Rose Enterprise Edition porque a pesar de ser una herramienta muy utilizada es un software propietario y no es multiplataforma. Características fundamentales que hicieron de Visual Paradigm la herramienta CASE seleccionada.

2.5. Lenguaje de programación PHP5

PHP (acrónimo de Hypertext Preprocessor) es un lenguaje del lado del servidor (esto significa que PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas Web dinámicas. Este puede ser incluido con facilidad dentro del código HTML y permite una serie de funcionalidades extraordinarias. Posee Licencia Open Source de Código Abierto que implica que el código fuente de PHP 5 es libre de ser descargado, puede ser utilizado, modificado y redistribuido sin coste alguno (5). Para el desarrollo del sistema se decidió usar PHP 5 como lenguaje de programación, por estar debidamente documentado, es un lenguaje

Capítulo 2: Selección y Descripción de Herramientas y Tecnologías

muy fácil de aprender y con su utilización se pueden programar de forma eficaz todas las funcionalidades que le dan solución al problema en cuestión, además de que permite la conectividad con la mayoría de los gestores de base de datos entre ellos PostgreSQL.

2.6. Sistema de Gestión de Bases de Datos (PostgreSQL v8.4).

PostgreSQL es un servidor de base de datos de objeto relacional libre, el cual está liberado bajo la licencia BSD, es decir, que puede ser utilizado, modificado y distribuido por todo el mundo, libre de cargo para cualquier propósito, sea comercial, privado o académico. Se considera uno de los sistemas gestores de bases de datos de código abierto más avanzada del mundo. PostgreSQL se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Windows, entre otros. Posee un soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, entre otros. Otra característica que fundamenta el uso de PostgreSQL es su módulo PostGIS que permite el trabajo con bases de datos espaciales (6).

2.6.1. PostGIS v1.5.3

PostGIS añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistemas de Información Geográfica. Se publica bajo la licencia pública general de GNU. Ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL y a juicio de muchos, es muy similar a la versión geográfica de Oracle. PostGIS ha sido certificado en 2006 por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas (6).

2.7. Servidor de Aplicaciones Web Apache2

Apache, es un servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. La licencia de Apache es una descendiente de la licencias BSD (Berkeley Software Distribution), lo que permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original. Es altamente configurable y de diseño modular, lo que permite ampliar las capacidades del servidor Web Apache de manera sencilla (5). Se escogió este servidor ya que permite el trabajo con PHP y otros lenguajes de script, además tiene todo el soporte que se necesita para tener páginas dinámicas. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Se puede configurar, para que ejecute un

Capítulo 2: Selección y Descripción de Herramientas y Tecnologías

determinado script cuando ocurra un error en concreto. Es una tecnología gratuita de código fuente abierta y multiplataforma

2.8. Servidor de Mapas MapServer v5.2.2

MapServer es un entorno de desarrollo de código abierto para la creación de aplicaciones SIG, este se ejecuta sobre plataformas Linux/Apache y Windows, soporta entre otros el formato vectorial PostGis y formatos ráster tales como JPG, PNG, GIF entre otros, además utilizando una API que proporciona MapScript se puede acceder a las funcionalidades de MapServer mediante lenguajes de programación como PHP, Java, C# y otros (6). El uso de este servidor de mapas permite la posibilidad de publicación de datos espaciales y aplicaciones de cartografía interactiva en la web.

2.9. Lenguaje de programación del lado del cliente (JavaScript)

Javascript es un lenguaje de programación web, que se utiliza para el desarrollo de páginas web, tiene funciones muy potentes que hace que una página web pueda ser dinámica, rápida y puede mejorar la interfaz de la misma. Al igual que otros lenguajes de programación web, es un lenguaje interpretado, lo que significa que el navegador web descarga el código de una página web y al encontrar en esta página código java script, lo interpreta y lo ejecuta en la computadora cliente (15).

2.9.1 Librería de JavaScript (Ext. JS v4.0)

ExtJS es una librería Javascript que permite construir aplicaciones complejas en Internet. Esta librería incluye:

- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open Source y comerciales.

Ext JS es muy flexible y permite realizar de manera rápida interfaces consistente, usa tecnologías tales como Ajax, DHTML y DOM. La comunidad que está detrás de esta herramienta es muy grande y la documentación cada vez es más extensa, cuenta con una API fácil de usar y su Licencia es Open Source, esta crea un balance en la arquitectura Cliente – Servidor, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo (5). Esta librería permite implementar de forma rápida y relativamente sencilla todas las interfaces con las que contará el módulo.

Capítulo 2: Selección y Descripción de Herramientas y Tecnologías

2.9.2 Librería de JavaScript (OpenLayers v2.10)

OpenLayers es una librería de JavaScript para visualizar los datos del mapa en la mayoría de los navegadores web modernos, sin dependencias de servidor. OpenLayers implementa una API de JavaScript para crear aplicaciones web geográficas, similares a las API de Google Maps, con una diferencia importante OpenLayers es un Software Libre. (16) Esta librería permite que se puedan dibujar sobre el mapa geometrías tales como puntos, líneas o polígonos y personalizarlas.

2.10. Framework de desarrollo (Symfony v1.4)

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Este framework además separa la lógica de negocio, la lógica de servidor y la presentación.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas como Unix, Linux, etc., además de Windows. A continuación se muestran algunas de sus características (7). Este

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Linux) independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Capítulo 2: Selección y Descripción de Herramientas y Tecnologías

2.11. Entorno de Desarrollo (NetBeans v7.1)

NetBeans es un entorno de desarrollo visual de código abierto. Su aprendizaje se ha convertido en fundamental para quienes están interesados en el desarrollo de aplicaciones multiplataforma. Está disponible para diversos sistemas operativos como Solaris, Windows, MacOS y GNU Linux. Su instalación y actualización es muy simple y una vez instalado se le pueden adicionar módulos que permiten extender sus funcionalidades. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (3). Dado a las características planteadas anteriormente, que permite el trabajo con el framework seleccionado (ya que permite generar los modelos de las clases de forma sencilla) y los lenguajes de programación (PHP, Ext JS) es ideal para el desarrollo del módulo a desarrollar ya que soportará toda la implementación del mismo.

2.12. Conclusiones

En el capítulo se expusieron las principales herramientas que fueron definidas para el desarrollo de la aplicación, estas herramientas poseen características encaminadas a facilitar el trabajo del desarrollador por lo que se cree que la elección está debidamente fundamentada. Se escogieron herramientas de código abierto para así obtener un producto totalmente libre y multiplataforma. Quedando definida RUP como metodología de desarrollo, UML como lenguaje de modelado, Visual Paradigm como herramienta CASE, los lenguajes de programación PHP 5 y Javascript, un SGBD potente como PostgreSQL, los servidores web y de mapas Apache 2.0 y MapServer respectivamente así como también las librerías Ext.JS y OpenLayer. De esta forma se aprovecharan todas las ventajas que proporciona cada una de estas herramientas y que tributarán al desarrollo de la aplicación.

Capítulo 3: Análisis y Diseño del Sistema

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Introducción

En este capítulo se hace una descripción de la solución propuesta a través del modelo de dominio representando los principales conceptos asociados al sistema. Se detallan los requisitos funcionales y no funcionales del sistema, casos de uso que se generan a partir de los requisitos funcionales y una explicación de cada uno de ellos a través de las descripciones textuales.

3.2. Modelo del Dominio

Basados en que no se cuenta con un cliente especializado en el análisis de los procesos del negocio que pueda definir de forma correcta los actores, los casos de uso y en general las fronteras del negocio, como solo se cuenta con unos pocos expertos por lo cual gran parte del conocimiento se obtiene mediante el estudio de sistemas similares al que se está desarrollando, y que el producto en desarrollo no está totalmente definido por lo que puede verse sujeto a cambios, se ha decidido recurrir al Modelo de Dominio. Este es un diagrama con los objetos reales relacionados con el proyecto, tiene como objetivo una mejor comprensión de los conceptos que utilizan los usuarios y solo se centra en la parte del negocio relacionada con el ámbito del proyecto.

Se llama **modelo del dominio** a la “representación visual de los conceptos u objetos del mundo real en un dominio de interés. Este modelo agrupa los conceptos de un dominio. Mecanismo fundamental para comprender el dominio del problema y para establecer conceptos comunes” (6). En él se representan los conceptos del dominio, sus características y las relaciones entre dichos conceptos. Es un diccionario visual del dominio del problema.

Capítulo 3: Análisis y Diseño del Sistema

3.2.1. Diagrama de clases del Modelo del Dominio

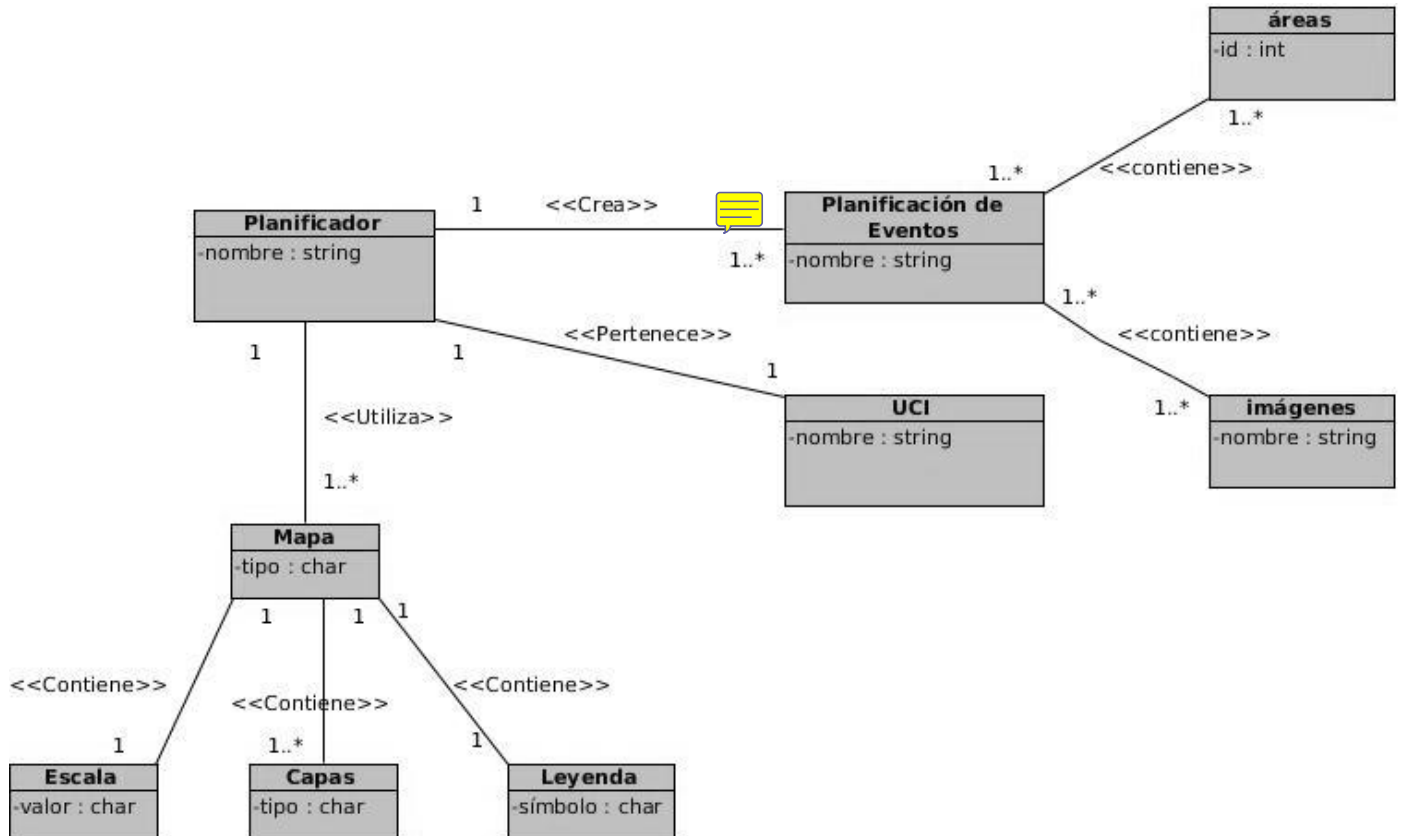


Imagen 4: Modelo de clases del dominio

3.2.2. Conceptos tratados en el modelo del dominio

Mapa: “Es una representación gráfica y métrica de una porción de territorio de acuerdo a los principios y métodos de la Cartografía”. (1)

Escala: “Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre”. (19)

Leyenda: Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa.

Capítulo 3: Análisis y Diseño del Sistema

Capa: Las capas son una forma de organizar la información temática para la elaboración de los SIG. Son transparencias colocadas a criterio del autor para facilitar la manipulación de la información. El sistema permite activar o no las capas disponibles.

Planificador: Persona que estará encargada de la realización de la planificación.




UCI: La universidad es la encargada de solicitar un servicio determinado utilizando un mapa.

Planificación de eventos: Contendrá todo lo relacionado con los eventos las áreas, las imágenes y la información referente a los mismos.

Áreas: Representa una persona o grupo de ellas señalando su ubicación en un determinado lugar y contendrá información referente a las mismas.

Imágenes: Representará una situación destacada dentro de un área.

3.3. Requisitos

Una etapa muy importante dentro del proceso de la Ingeniería de *Software*, es la Ingeniería de Requisitos, donde se lleva a cabo el proceso de descubrir, analizar y verificar los servicios y restricciones del sistema de *software* que se de  producir. Esta no es más que el conjunto de actividades y tareas del proceso de desarrollo de **sistemas software**.

Los requisitos no son más que una “Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”. (8)

3.3.1. Requisitos funcionales

RF1.	El sistema debe permitir adicionar eventos.
RF2.	El sistema debe permitir eliminar eventos.
RF3.	El sistema debe permitir crear un área.

Capítulo 3: Análisis y Diseño del Sistema

RF4.	El sistema debe permitir modificar un área.
RF5.	El sistema debe permitir eliminar un área.
RF6.	El sistema debe permitir insertar imágenes.
RF7.	El sistema debe permitir mover imágenes.
RF8.	El sistema debe permitir eliminar imágenes.
RF9.	El sistema debe permitir mostrar información.
RF10.	El sistema debe permitir publicar las planificaciones de los eventos.
RF11.	El sistema debe permitir visualizar la planificación publicada de los eventos.

Tabla 1: Requisitos funcionales

Descripción de los requisitos funcionales

A continuación se describen los requisitos funcionales más importantes del Módulo de Georeferenciación Eventual para el SIG-UCI. Sus correctas descripciones son vitales, puesto que ayudan a comprender el funcionamiento total del sistema.

RF1. Adicionar eventos.

Este requerimiento permite: que el usuario encargado de la planificación de los eventos pueda añadir una nueva capa al sistema, la cual representará el evento que se quiera tratar. El sistema permitirá que dicha capa pueda ser modificada con líneas, puntos, polígonos, imágenes e información que harán referencia a las características de los eventos a representar.

RF3 Crear áreas.

Capítulo 3: Análisis y Diseño del Sistema

Este requerimiento permite: que luego de ser añadido un nuevo evento el usuario pueda utilizar las opciones que se le brindan para el trazado de un área determinada ya sea mediante puntos, líneas, polígonos o una combinación de estas.

RF4 Modificar áreas.

Este requerimiento permite: modificar un área determinada en el momento en que fue creada mediante el cambio de forma, el aumento o disminución del tamaño para determinar el espacio que ocupará sobre una superficie determinada.

RF6 Insertar imágenes.

Este requerimiento permite: que una vez que el usuario haya definido un área determinada, el sistema le brinde la posibilidad de adicionar imágenes que tengan un significado lógico y que simulen lo que se quiere contener dentro del área en cuestión.

RF6 Mostrar Información.

Este requerimiento permite: que el usuario pueda visualizar la información referente a un área determinada una vez que esta haya sido adicionada ingresando al sistema el identificador de la misma.

3.3.2 Requisitos no funcionales

Los requisitos no funcionales forman una parte significativa del producto. Estos constituyen rasgos fundamentales que hacen que el sistema sea usable, rápido, confiable y amigable para los usuarios. A continuación se exponen los utilizados para el desarrollo del sistema que dará solución al problema que dio inicio a la investigación. Todos los requisitos no funcionales están basados en el documento Especificación de requisitos de software v2.0 del proyecto SIGUCI.

➤ Apariencia y Usabilidad

- El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. La estructura de las interfaces será clara, las funcionalidades principales estarán separadas en diferentes Tabs para así mantener una mejor organización de lo que se va a mostrar.

Capítulo 3: Análisis y Diseño del Sistema

➤ Eficiencia

- El tiempo de respuesta será directamente proporcional a la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento y viceversa. La actualización también dependerá de la cantidad de información que se tenga que procesar.

➤ Fiabilidad

- La información manejada por el sistema estará protegida de acceso no autorizado. Los dispositivos o mecanismos utilizados para lograr la seguridad no retrasarán a los usuarios para obtener los datos deseados en un momento dado.

➤ Hardware

Para determinar los requerimientos de Hardware se partió de los requerimientos mínimos que necesita el sistema para su ejecución:

Para las máquinas clientes:

- Se requiere tengan tarjeta de red.
- Al menos 128 MB de memoria RAM.
- Se requiere al menos 100 MB de disco duro.
- Procesador 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- El Servidor de Mapas tenga como mínimo 700 MB de RAM y 2 GB de disco duro.
- El Servidor de BD tenga como mínimo 2 GB de RAM y 10 GB de disco duro.
- Procesador 3 GHz como mínimo.

➤ Restricciones de diseño.

Capítulo 3: Análisis y Diseño del Sistema

Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.

Para las PCs cliente.

- Un Navegador como Mozilla Firefox, Safari u otro navegador que cumpla con los estándares de W3C (World Wide Web) compañía que desarrolla estos estándares.

Para los servidores.

- Sistemas operativos GNU/Linux.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5.
- PostgreSQLv8.4 como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.

➤ Requisitos de Licencia

- De acuerdo a los tipos de licencias de los componentes y herramientas que se proponen a utilizar para el desarrollo del módulo se puede catalogar legalmente esta arquitectura de modelo libre. Permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

3.4. Casos de uso del sistema

Un caso de uso es la forma en que interactúa un usuario con alguna de las funcionalidades de un sistema. Todo sistema de software ofrece a su entorno una serie de servicios, un caso de uso expresa la forma de cómo alguien externo a un sistema lo usa. Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema (18).

“La parte más difícil de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con gente, máquinas, y otros sistemas. Ninguna otra parte del trabajo afecta tanto al sistema si es hecha mal. Ninguna es tan difícil de corregir más adelante... Entonces, la tarea más importante que

Capítulo 3: Análisis y Diseño del Sistema

el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto". (11)

3.4.1 Diagrama de casos de uso del sistema

Un diagrama de Casos de Uso del Sistema describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan (13).

Un actor es un rol que cumple un usuario, puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema (14).

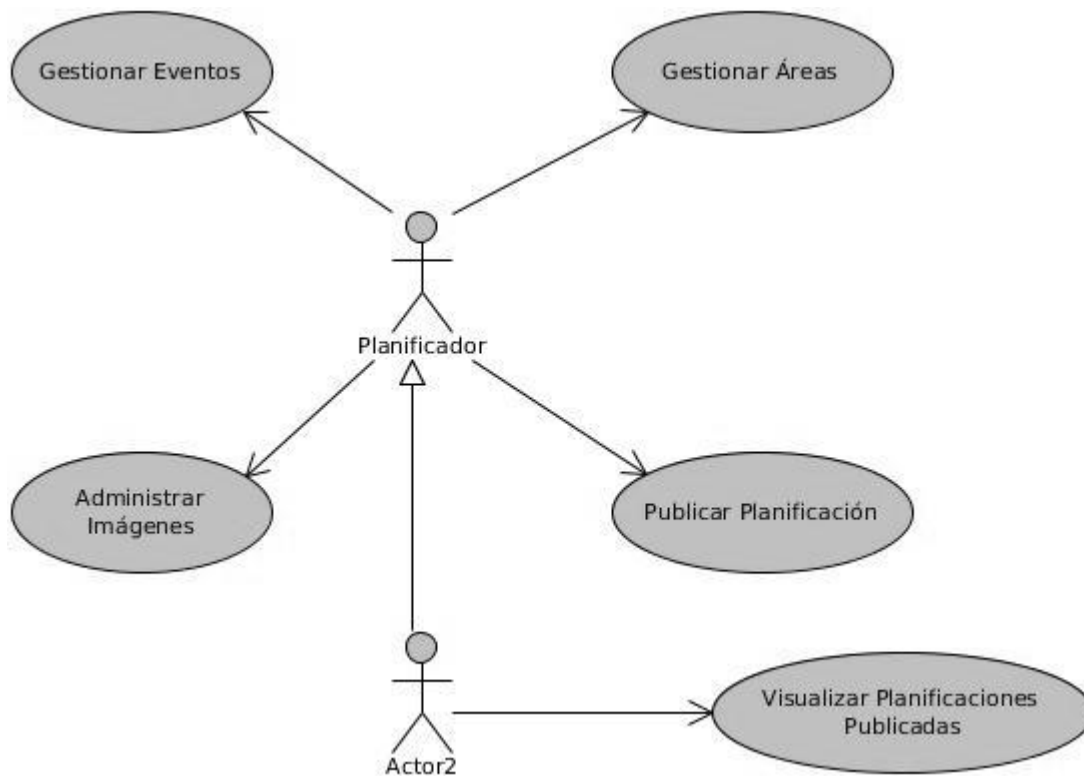


Imagen 5: Diagrama de Casos de Uso

Objetivo	Gestionar un área
Actores	Usuario :(Inicia).

Capítulo 3: Análisis y Diseño del Sistema

Resumen	El caso de uso se inicia cuando el usuario necesita representar sobre el mapa un área determinada y proporcionarle una serie de elementos para que represente un conjunto de participantes en un evento. El caso de uso concluye una vez que el área debidamente dibujada y modificada es añadida a la base de datos.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Ninguna.	
Postcondiciones		
Flujo de eventos		
Flujo básico: Gestionar áreas.		
	Actor	Sistema
1.	Selecciona la opción "GeoreferenciarEventos" en la barra de menú.	
2.		Muestra la interfaz "Georeferenciareventos".
3.	Puede seleccionar las opciones para adicionar, modificar o eliminar un área.	

Capítulo 3: Análisis y Diseño del Sistema

4.		<p>Habilita las opciones de Gestionar áreas.</p> <ul style="list-style-type: none"> • Adicionar área. Ver Sección 1: “Adicionar área”. • Modificar área. Ver Sección 2: “Modificar área”. • Eliminar área. Ver Sección 3: “Eliminar área”. <p>Termina el caso de uso.</p>
Sección 1: “Adicionar área”		
Flujo Básico: Adicionar área.		
	Actor	Sistema
5.	Selecciona la opción Pintar área.	
6.		Muestra los distintos tipos de áreas a pintar (líneas, puntos o polígonos).
7.	Selecciona el tipo de área y comienza a pintar oprimiendo el botón con el nombre del área.	
8.		Se muestra en el mapa el área dibujada.
Sección 2: “Modificar área”		

Capítulo 3: Análisis y Diseño del Sistema

Flujo Básico: Modificar área.			
	Actor	Sistema	
9.	Selecciona el tipo de área que desea modificar y procede a dar clic sobre ella y modificarla.		
10.		Se muestra el área modificada.	
Sección 3: “ Eliminar área”			
Flujo Básico: Eliminar área.			
	Actor	Sistema	
11.	Entra el identificador del área que se desea eliminar.		
12.		Busca en la base de datos y elimina el área. Muestra un mensaje notificando la correcta eliminación.	
Relaciones		CU Incluidos	Ninguno.
		CU Extendidos	Ninguno.
Requisitos no funcionales		Ninguno.	
Asuntos pendientes		No procede.	

Tabla 2 Descripción del CU Gestionar áreas

3.5. Conclusiones

La especificación del Modelo de Dominio arrojó como resultado que todo el interesado pueda adquirir un entendimiento mínimo del contexto en que se emplaza el Módulo de Georeferenciación Eventual y conocer

Capítulo 3: Análisis y Diseño del Sistema

el significado de los principales conceptos asociados al mismo. Se definieron una serie de requisitos funcionales los cuales conforman la base de lo que será implementado y mediante su correcta descripción quedó definido de qué forma se hará.

Capítulo 4: Implementación y Prueba

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción.

En el siguiente capítulo se explica la arquitectura empleada y el conjunto de patrones de diseño usados para garantizar la viabilidad y calidad requerida del sistema a construir. De igual manera se analiza la fase de implementación a partir de los resultados del diseño, describiendo el estado actual del sistema en término de paquetes.

4.2 Descripción de la Arquitectura.

La Arquitectura de Software es un diseño de alto nivel de la estructura de un sistema. En la misma se describen los patrones y diagramas más relevantes estableciendo un modelo o línea base de desarrollo. De esta forma se crea un sistema de abstracciones coherentes que proporciona el marco de referencia necesario y erige las pautas a seguir en la construcción del sistema. (8).

Para la realización del módulo se definió utilizar el estilo de Llamada y Retorno, que permite construir una estructura de programa relativamente fácil de modificar y ajustar a escala. Dentro de este, el patrón de diseño Modelo Vista Controlador (MVC), en estructura en capas. Donde todo el proceso está dividido en 3 partes, el Modelo, la Vista y el Controlador. Para la capa de acceso a datos se utiliza el PHP 5 con todas las bondades que el mismo brinda, así como postgre SQL para la capa de datos y Ext.js para la vista.

El patrón MVC se centra en la separación del modelo y la vista, mientras que el controlador es el encargado de relacionarlos. Su principal característica es aislar la vista del modelo. El patrón MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de la aplicación, es decir, se tratan como entidades separadas. Esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

La Vista será responsable de mostrar, mediante el uso de interfaces gráficas, la información almacenada en el modelo. Además será la encargada de recibir información del usuario mediante acciones efectuadas por el mismo a través de dichas interfaces.

El Controlador especificará todas las reglas de negocio que permitirán ejecutar la totalidad de las operaciones que son ordenadas desde los objetos gráficos que se han definido previamente en el

Capítulo 4: Implementación y Prueba

componente interfaz. Para ello se debe procesar la información que toma del paquete Vista para devolverle finalmente una respuesta, de modo que el usuario sea capaz de visualizar el resultado de la operación que ha ordenado. Además, se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones. El Modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes. También realiza el encapsulamiento de los datos almacenados en objetos accesibles por los otros dos paquetes del sistema: Controlador y Vista. (6)

4.2.1. Ventajas que presenta este estilo

- Permite tener un completo control sobre el comportamiento de una aplicación.
- Soporte de vistas múltiples, es decir, dado que la vista se encuentra aislada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- Agregar nuevos tipos de vistas no afecta al modelo.
- Permite un diseño modular, posibilitando que los diseñadores y los desarrolladores trabajen conjuntamente.

4.2.2 Patrones de diseño.

Los patrones de diseño proporcionan un esquema para refinar los subsistemas y las relaciones entre los componentes de un sistema. Conjuntamente componen soluciones exitosas a problemas frecuentes que se presentan durante el diseño de una aplicación. Impiden cometer errores, son soluciones concretas a problemas frecuentes que se presentan, están basados en la experiencia acumulada de expertos de desarrollo de software (17).

A continuación serán expuestos y descritos los patrones que fueron utilizados en el desarrollo del Módulo, tanto los que rigieron el diseño de la aplicación (GOF) como los Patrones Generales de Software para la Asignación de Responsabilidades (GRASP).

Patrones Creacionales:

Capítulo 4: Implementación y Prueba

Ofrecen soporte a una de las tareas más comunes dentro de la programación orientada a objetos: la instanciación. Estos ayudan a crear un sistema independiente de cómo se crean, se componen y se representan sus objetos.

➤ Singleton(Instancia única)

Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a esta instancia.

Beneficios:

- Acceso controlado a una única instancia, es decir, encapsula su única instancia.
- Permite el perfeccionamiento de las operaciones y de la representación.
- Facilita cambiar, atender y permitir más de una instancia de la clase Singleton.
- Usar este patrón es más flexible que las clases operacionales.

Se evidencia el uso del patrón cuando en la clase Actions es creada una única instancia de la clase EventoTable para acceder a todos los métodos que la misma posee.

Patrones de diseño GRASP:

Describen los principios esenciales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

➤ Patrón Controlador.

El propósito de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Beneficios:

- Mayor potencial de los componentes reutilizables.

Capítulo 4: Implementación y Prueba

- Razonamiento sobre el estado de los casos de uso (Asegurar que la secuencia es válida y razonar sobre el estado actual).

El uso del patrón se evidencia en la clase Actions la cual controla todas las peticiones que realiza el sistema, esta constituye el único punto de entrada de la aplicación.

➤ Patrón Bajo Acoplamiento.

Este patrón especifica cómo dar soporte a una dependencia escasa y a un aumento de la reutilización, enfocándose en asignar una responsabilidad para mantener bajo acoplamiento. El bajo acoplamiento estimula la asignación de responsabilidades de forma tal que la inclusión de éstas no incremente el acoplamiento, creando clases más independientes y con mayor resistencia al impacto de los campos, que aumentan la productividad y la posibilidad de reutilización.

El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios. Es válido precisar que este patrón no puede verse de forma independiente a los patrones Experto y Alta Cohesión, sino más bien incluirse como otro de los principios del diseño que influyen de forma determinante a la hora de la asignación de responsabilidades.

Beneficios:

- No afectan los cambios en otros componentes.
- Conveniente para reutilizar.
- Fácil de entender de manera aislada.

El uso del patrón se evidencia puesto que el modelo contiene las clases que implementan la lógica del negocio y de acceso a datos, las cuales no tienen asociaciones con las de la vista o el controlador, logrando que la dependencia sea baja.

➤ Patrón Alta Cohesión.

El objetivo de este patrón es asignar responsabilidades de tal forma que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Capítulo 4: Implementación y Prueba

Beneficios:

- Incrementa la claridad y facilidad de comprensión del diseño.
- Simplifica el mantenimiento.
- Soporta bajo acoplamiento. Una mala cohesión lleva a mal acoplamiento y viceversa.
- El grano fino de funcionalidad altamente relacionada incrementa la reutilización.

La evidencia del patrón la demuestra el hecho de que la clase Actions permite la organización del trabajo mediante la asignación de responsabilidades.

➤ Patrón Experto

Este patrón plantea asignar una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad. El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos.

Beneficios:

- Garantizar el encapsulamiento de la información.
- Facilita el bajo acoplamiento en las aplicaciones.

La evidencia del patrón lo demuestra el hecho de que La clase EventoTable del modelo contiene todas las funcionalidades relacionadas con los eventos permitiendo el acceso a los datos de los mismos.

➤ Patrón Creador

Este patrón es el encargado de asignar responsabilidades para la creación de objetos.

La evidencia de este patrón se muestra en las clases Table cuando se crean objetos de tipo Query para posibilitar el acceso a la información almacenada en la base de datos.

Capítulo 4: Implementación y Prueba

4.2.3. Diagrama de clases del diseño

El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso y al mismo tiempo constituye una abstracción del modelo de implementación. Dentro de los artefactos que genera este modelo, se encuentra el Diagrama de Clases del Diseño, en el cual se muestran un conjunto de clases, interfaces y colaboraciones, así como las relaciones entre sí. Se realiza cada diagrama por caso de uso identificado, estos diagramas son importantes pues visualizan, especifican y documentan modelos estructurales, lográndose una muestra amplia y confiable del sistema previo a su implementación (12).

A continuación se muestra el diagrama del caso de uso Gestionar Áreas, diseñado bajo la arquitectura MVC. Posteriormente se realiza una descripción de las clases que se representan para un mejor entendimiento de sus funciones.

Capítulo 4: Implementación y Prueba

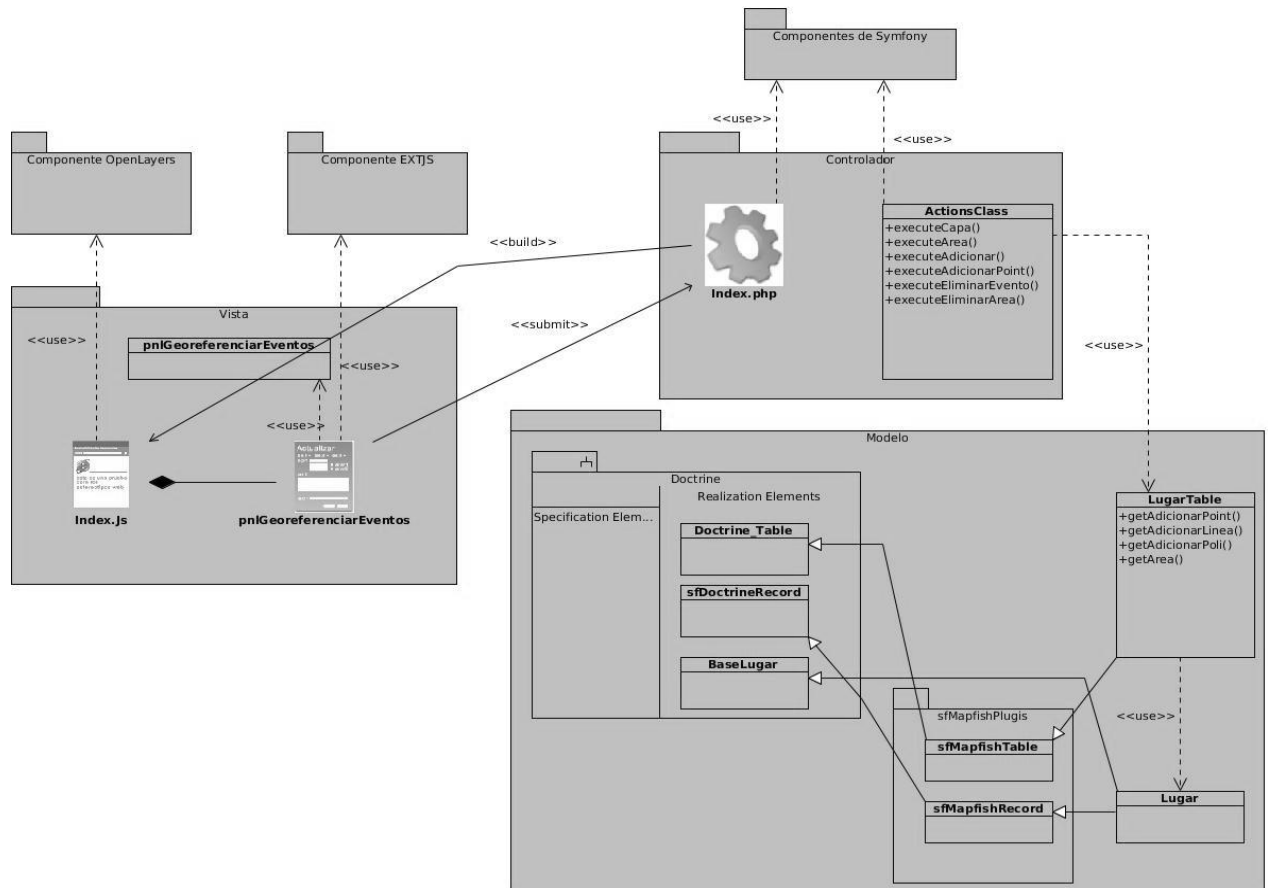


Imagen 6: Diagrama de clases del diseño gestionar áreas

Descripción de clases

Cada uno de los diagramas de clases del diseño que emplean el framework Symfony poseen clases que son comunes para la mayoría de ellos, tanto las que son JavaScript como las que son PHP. A continuación serán descritas las clases fundamentales del diagrama y las relaciones de comunicación que se realizan entre ellas.

Clase Index.js: clase encargada de construir todos los componentes visuales del formulario, además recibe y procesa las peticiones del usuario.

Capítulo 4: Implementación y Prueba

Clase LugarTable: es la clase encargada de establecer la conexión con el servidor de base de datos para obtener los datos a procesar.

Clase Actions: tiene como propósito controlar la realización del CU en sí, recibe las peticiones realizadas por el cliente, gestiona las mismas y envía la respuesta a la Vista.

Clase Index.php: Es el controlador frontal, todas las peticiones son manipuladas por el controlador frontal que es el punto de entrada de la aplicación.

Clase pnlGeoreferenciarEventos: controla la información de la vista, permite hacer validaciones, por ejemplo valida que no existan campos vacíos.

Clase pnlGeoreferenciarEventos (formulario): interactúa con el usuario, recibe y muestra información.

4.3. Diseño de la base de datos

El diseño de la base de datos brinda persistencia al modelo descrito. Es capaz de reflejar la estructura del problema en el mundo real.

4.3.1 Modelo de datos.

En el Modelo de Datos, el analista introduce el diseño conceptual o representación de la estructura de información de la aplicación, siguiendo, paso a paso, la metodología Entidad-Relación. Aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos (20).

Capítulo 4: Implementación y Prueba

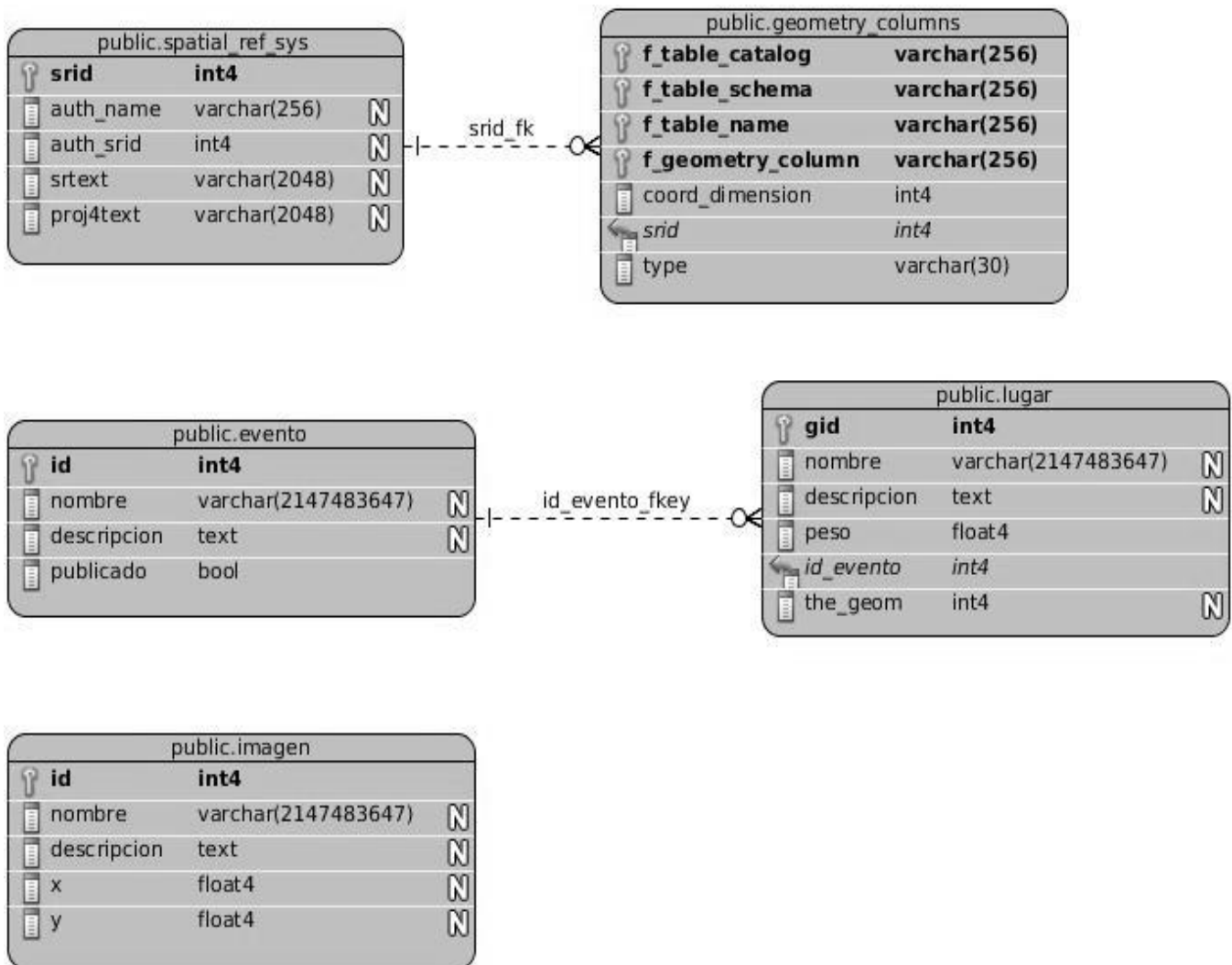


Imagen 7: Diagrama Entidad-Relación

4.4. Distribución física del sistema

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP. (22)

Capítulo 4: Implementación y Prueba

El diagrama de despliegue correspondiente al SIG-UCI al cual se le desarrolla el Módulo de Georeferenciación Eventual, como se muestra a continuación consta con una computadora cliente, en la cual los usuarios pueden visualizar la planificación interactuando con la aplicación que se encuentra en el servidor. Este servidor es el encargado de responder las peticiones de las computadoras clientes y a su vez está conectado al servidor de mapas, el cual es el encargado de administrar los mapas necesarios para el funcionamiento correcto de la aplicación. Además está conectado a un servidor que provee los datos geográficos para visualizar en el mapa la información.

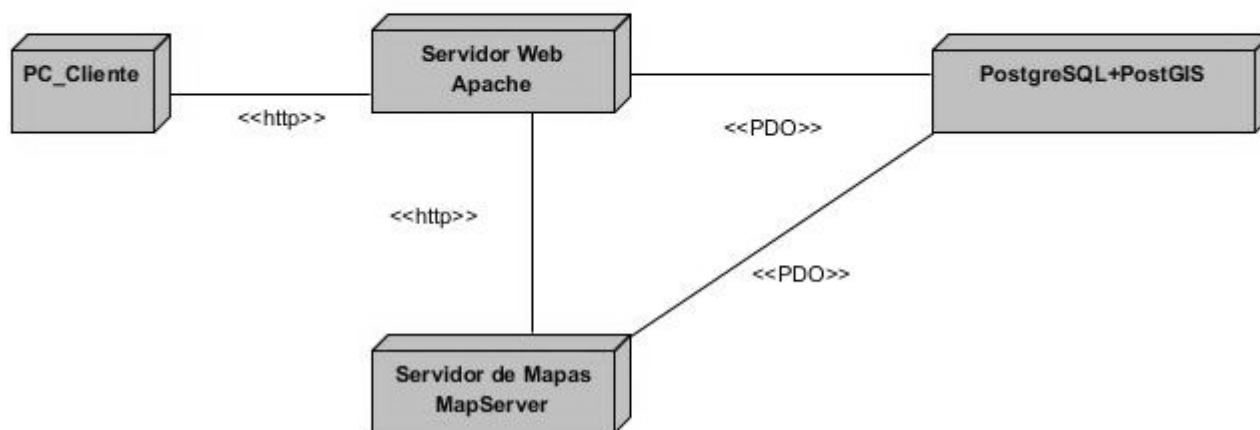


Imagen 8: Diagrama de despliegue del Módulo de Georeferenciación Eventual

Protocolo HTTP

El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW), HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (cliente, servidor, proxy) para comunicarse. Se utiliza para el acceso de los clientes web a los servicios que brinda la aplicación (12).

Capítulo 4: Implementación y Prueba

PDO

La extensión Objetos de Datos de PHP (PDO por sus siglas en inglés) define una interfaz ligera para poder acceder a bases de datos en PHP. PDO proporciona una capa de abstracción de acceso a datos, lo que significa que, independientemente de la base de datos que se esté utilizando, se usan las mismas funciones para realizar consultas y obtener datos (21).

4.5 Modelo de implementación

El modelo de implementación es una visión general del módulo donde se describen los elementos físicos del mismo y sus relaciones. Está constituido por una colección de componentes que representan todos los tipos de elementos de software que forman parte de la aplicación.

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables, ilustran las piezas del software, controladores embebidos, etc. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema, es decir para describir la vista de implementación estática de un sistema. Los diagramas de componentes se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones pero un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema (5).

Capítulo 4: Implementación y Prueba

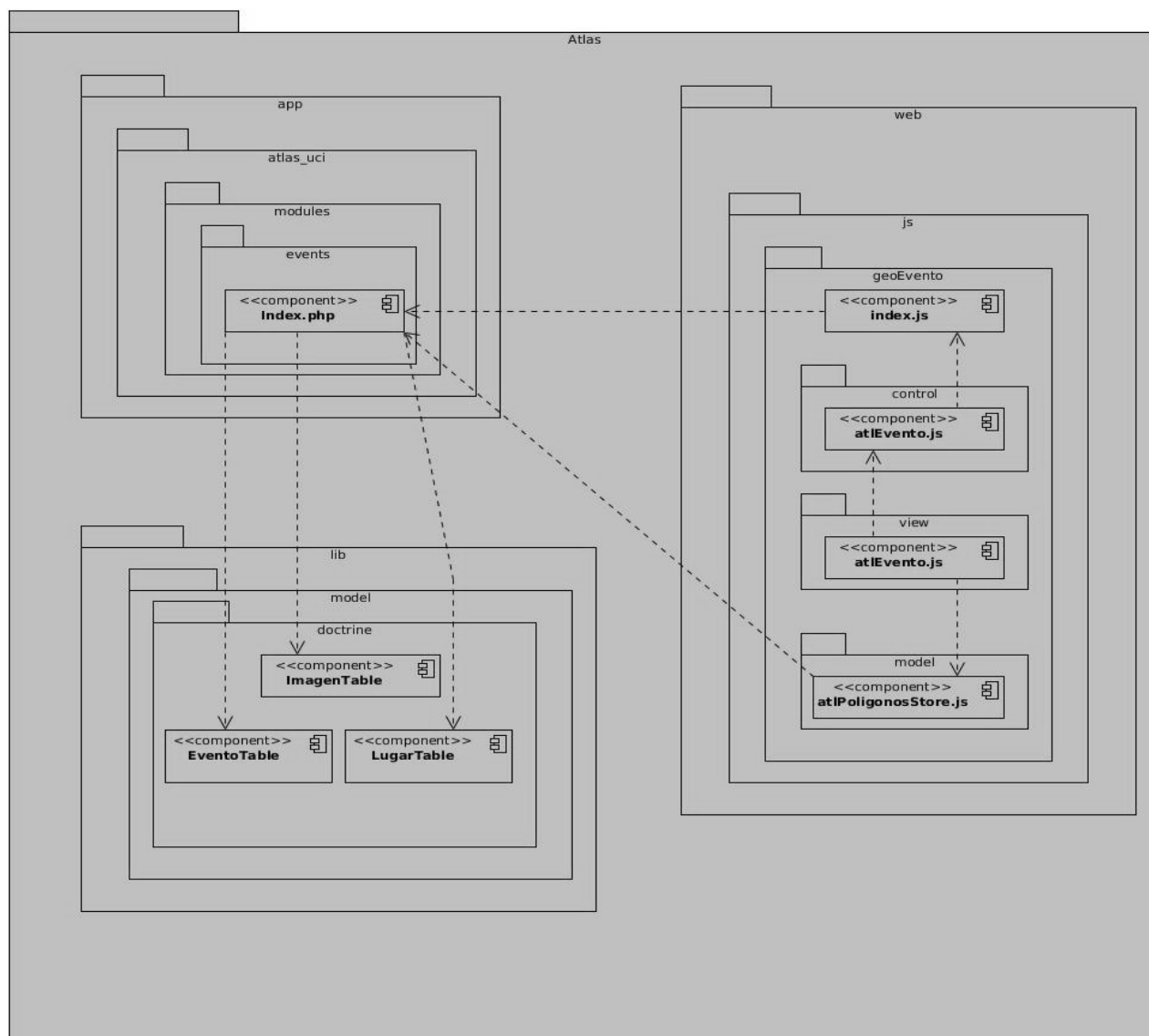


Imagen 9: Diagrama de componentes del Módulo de Georeferenciación Eventual

4.6. Pruebas realizadas al sistema

El único instrumento adecuado para determinar el grado de la calidad de un producto de software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Se usan casos de prueba, especificados de forma estructurada mediante Técnicas de

Capítulo 4: Implementación y Prueba



prueba. El proceso de pruebas, sus objetivos y los métodos y técnicas usados se describen en el plan de prueba. (4)

Con el objetivo de mostrar el correcto funcionamiento de la aplicación y comprobar que las operaciones internas se ajustan a las especificaciones planteadas se realizaron pruebas de caja negra. Las pruebas de caja negra o pruebas de entrada/salida son las que no toman en cuenta el código, el que lo prueba no sabe cómo está estructurado por dentro el programa o bien no necesita saber nada de programación, solo necesita saber cuáles pueden ser las posibles entradas sin necesidad de entender cómo se deben obtener las salidas. Estas pruebas tratan de encontrar errores en la interfaz mientras se está usando.

Los casos de prueba de caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- Las entradas se aceptan de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

La técnica a utilizada dentro de las pruebas de caja negra es partición de equivalencia, la cual permite examinar los valores validos e inválidos de las entradas del software, esta técnica va dirigida a la definición de casos de pruebas que descubran clases de errores para de esta forma reducir el número de casos de pruebas a desarrollar.

Caso de prueba: Gestionar eventos.

Descripción general: El caso de uso se inicia cuando el usuario necesita destacar un objetivo dentro de un área determinada y para ello selecciona una de las imágenes posibles a insertar.

Condiciones de ejecución: Ninguna.

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo Central
----------------------	-------------------------	---------------------------------	---------------

Capítulo 4: Implementación y Prueba

SC1: Adicionar evento.	EC 1.1: Adicionar eventos correctamente.	El usuario entrará el nombre y la descripción del evento al sistema y se mostrará la información en el control de capas.	<ol style="list-style-type: none"> 1. Entrar el nombre y la descripción del evento. 2. Clic en el botón "RegistrarEvento". 3. Se carga el nombre del evento en el control de capas.
SC2: Eliminar evento.	EC 3.1: Eliminar un evento.	El usuario entra al sistema el nombre del evento que desea eliminar y presiona el botón "DeleteEv"	<ol style="list-style-type: none"> 4. Se le entra el sistema el nombre del evento. 5. Clic en el botón "DeleteEv". 6. Se muestra un mensaje notificando la correcta eliminación.

Tabla 3: Secciones a probar en el CU Gestionar evento.

No.	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Debe entrar un nombre.
	Descripcion	Campo de texto	No	Debe entrar una descripción.

Tabla 4: Descripción de variables

Capítulo 4: Implementación y Prueba

Id del Escenario	Escenario	1	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Adicionar evento.	V "Evento"	Se muestra el nombre del evento en el control de capas.	Satisfactorio
EC 2.1	Eliminar evento	V "Evento"	Se elimina el evento de la base de datos y del control de capas.	Satisfactorio

Tabla 5: Matriz de datos del CU Gestionar eventos

Se realizaron tres iteraciones de pruebas, en la primera de ellas se arrojaron las siguientes no conformidades.

- La interfaz del sistema se encontraba un poco cargada de elementos.
- La información referente a los eventos no se mostraba de manera correcta.
- Los eventos no se eliminaban correctamente (no se eliminaban las áreas asociadas a ellos).

En la segunda iteración aparece una nueva no conformidad referente al caso de uso gestionar imágenes.

- Las imágenes no se guardan en la base de datos.

Las no conformidades fueron resueltas y una tercera iteración demostró que el software arroja todos los resultados de manera satisfactoria.

4.7. Conclusiones Parciales

En el presente capítulo se adoptó la arquitectura definida por el SIG-UCI, facilitando la reutilización de componentes de software ya implementados para un aumentando en la rapidez del desarrollo del módulo. Se facilita la comprensión y el mantenimiento del código debido a los patrones de arquitectura y diseño

Capítulo 4: Implementación y Prueba

definidos. Los artefactos y diagramas generados referentes al flujo de trabajo de implementación favorecieron la comprensión del proceso de desarrollo. La selección del método de prueba de caja negra y la técnica partición de equivalencia permitieron detectar posibles no conformidades que posee el módulo desarrollado y verificar que la aplicación funciona correctamente.

Conclusiones

CONCLUSIONES

Producto de la investigación desarrollada y los resultados alcanzados se llegan a las siguientes conclusiones:

- Partiendo del análisis de la problemática planteada y luego de ser analizadas soluciones homólogas, se obtiene una herramienta que cumple con los objetivos trazados. Tanto la metodología, como las herramientas que fueron seleccionadas para la construcción del módulo aportaron los elementos necesarios para la correcta implementación, así como también generaron una serie de artefactos necesarios para la comprensión del sistema y soportaron todo el ciclo de vida del producto.
- Se logró finalizar con el desarrollo del módulo propuesto, verificando su correcto funcionamiento mediante el método de prueba de caja negra. La correcta implementación del módulo y la documentación generada que sirve para ser utilizada como material de consulta en futuras investigaciones, constituyen los principales aportes del presente trabajo de diploma, además de que se le brinda al SIG-UCI una herramienta que será capaz de representar la distribución de las personas que estén participando en los diferentes eventos que tienen lugar dentro de la universidad.

Recomendaciones

RECOMENDACIONES

- Se recomienda que se amplíe la gama de imágenes que un usuario puede adicionarle a un evento, que no se limite solo a insertar imágenes que se encuentran guardadas en la base de datos sino que se pueda insertar cualquiera sin importar el lugar donde se encuentre guardada.

Bibliografía

REFERENCIAS BIBLIOGRÁFICAS

Trabajos citados

1. ecured. [Online] [Cited: 4 10, 2011.] <http://www.ecured.cu/index.php/Mapa..>
2. ecured. [Online] 2009. [Cited: 3 27, 2012.] http://www.ecured.cu/index.php/Sistema_de_Información_Geográfica.
3. Netbeans. [Online] [Cited: 2 16, 2011.] <http://netbeans.org/features/index.html.>
4. PRUEBASDESFTWARE. [Online] 2005. [Cited: 3 27, 2012.] <http://pruebasdesoftware.com/laspruebasdesoftware.htm.>
5. **Gorgoy, Osvaldo Garcia.** *Sistema de Información Geográfica para el control de los recursos hidráulicos cubanos.* La Habana : s.n., 2011.
6. **Matos, Yeisy Legrá.** *Módulo de administración del Sistema de Información Geográfica para el transporte obrero de la Universidad de las Ciencias Informáticas.* La Habana : s.n., 2011.
7. **Potencier, Fabien and Zaninotto, Francois.** Librosweb. *Symfony 1.4, la guía definitiva.* [Online] [Cited: 3 27, 2012.] <http://www.librosweb.es.>
8. **Pressman, Roger S. 2005.** *Ingeniería del Software un enfoque práctico.* La Habana: Félix Varela, 2005.
9. **Ramírez, Leiber Fornaris and León Companioni, Alain.** *Sistema para la representación geográfica de objetivos hidrográficos de precipitaciones.* La Habana : s.n., 2012.
10. **Zapata, Yasmany Pérez.** *Estrategia para la replicación de datos espaciales en Sistemas de Información Geográfica.* La Habana : s.n., 2008.
11. **Ledea, Lilianne Martínez.** *Especificación de requisitos de software v2.0.* La Habana : s.n., 2012.

Bibliografía

12. **Bosque, Est. Lianet Maldonado.** *Modelo de diseño.* La Habana : s.n., 2011.
13. **Ortiz, Jackeline Fidalgo y Rios Morales, Fredy Hector.** *Desarrollo de un componente se software para importar reportes desde documentos Excel a Sistemas Gestores de Bases de Datos en el CEINPET.* La Habana : s.n., 2010.
14. **Rosales, Carlos Luis Serrano y Rodríguez Vásquez, Solangel.** *Desarrollo de Biblioteca de Métodos Numéricos (BMN), referente a Sistemas de Ecuaciones Lineales, Sistemas de Gran Dimensión y Poco Densos e Integración Numérica.* La Habana : s.n., 2008.
15. **Bermúdez, Carlos.** Código de Programación. [En línea] Código de Programación, 30 de mayo de 2010. [Citado el: 1 de diciembre de 2012.] <http://www.dgtallika.com/2010/05/definición-de-hoy-javascript/>.
16. **OpenLayers.** OpenLayers. *OpenLayers.* [Online] OpenLayers, 2012. <http://openlayers.org/>.
17. **Ricardo Armando Machorro.** Los Patrones como un Medio del Diseño Orientado a Objetos. [En línea] [Citado el: 17 de marzo de 2012.] www.revistaupiicsa.20m.com/Emilia/RevMayAgo04/Machorro1.pdf.
18. **mastermagazine.** [Online] [Cited: 5 28, 2013.] <http://www.mastermagazine.info/termino/4184.php>.
19. **ecured.** ecured. [En línea] [Citado el: 10 de 4 de 2011.] <http://www.ecured.cu/index.php/Escala>.
20. **—.** ecured. [En línea] [Citado el: 2 de 5 de 2012.] http://www.ecured.cu/index.php/SNAP#Modelo_de_Datos.
21. **php.net.** [En línea] 3 de 5 de 2013. [Citado el: 8 de 5 de 2013.] <http://php.net/manual/es/intro.pdo.ph>

Bibliografía

22. **ecured.** ecured. [En línea] [Citado el: 12 de 5 de 2012.]
[http://www.ecured.cu/index.php/Diagrama_de_despliegue.](http://www.ecured.cu/index.php/Diagrama_de_despliegue)
23. **definición.** *definición.de.* [En línea] 2008. [Citado el: 20 de noviembre de 2011.]
<http://definición.de/cartografía/.,2008>
24. **SIG-fam, grupo.** Sistemas de Información Geográfica FAM. [Online] grupo SIG-fam, 2010.
[siffam.com.ar/content/view/128/2/.](http://siffam.com.ar/content/view/128/2/)

Bibliografía

Trabajos consultados

Alvarez, Miguel Angel. 2004. Desarrolloweb.com. [Online] noviembre 2004. <http://www.desarrolloweb.com/articulos/1696.php>.

Antioquía, Universidad de. 2011. Banco de objetos de aprendizaje y de información. [En línea] 2011. [Citado el: 1 de diciembre de 2012.]

Astudillo, Marcello Visconti y Hernán. 2010. *Fundamentos de Ingeniería de Software*. s.l.: Universidad Técnica Federico Santa María, 2010.

Alborum. alborum. *alborum*. [En línea] [Citado el: 18 de abril de 2012.] <http://www.alborum.es/index.php/envien/noticias/195-el-gasto-en-España-por-direcciones-postales-imprecisas>.

Bermúdez, Carlos. 2010. Código de Programación. [En línea] Código de Programación, 30 de mayo de 2010. [Citado el: 1 de diciembre de 2012.] <http://www.dgtallika.com/2010/05/definición-de-hoy-javascript/>.

Carles, Richard Fifer. 1992. Geoinfo. *Geoinfo*. [En línea] Geoinfo, 1992. [Citado el: 30 de noviembre de 2011.] <http://www.geoinfo-int.com>.

CHAPARRO, JOAQUÍN RODRÍGUEZ. 2009. SISTEMAS DE INFORMACIÓN GEOGRÁFICA. *SISTEMAS DE INFORMACIÓN GEOGRÁFICA*. [En línea] 2009. [Citado el: 30 de noviembre de 2011.] www.mappinginteractivo.com.

Ciberaula. 2006. Patrones de Diseño en aplicaciones web con Java J2EE. *Ciberaula-Java*. [En línea] 2006. [Citado el: 10 de Abril de 2010.] http://java.ciberaula.com/artículo/diseño_patrones_j2ee/.

Connexions. 2005. Connexions . *Los Sistemas de Gestión de Bases de Datos son software especializados*. [En línea] 2005. <http://cnx.org/>.

Bibliografía

COLOMBIA., UNIVERSIDAD COOPERATIVA DE. 2012. PATRONES DE DISEÑO Y FRAMEWORKS. [Online] 2012. [Cited: diciembre 3, 2012.] <http://ingenieriasw2.blogspot.com/p/patrones-de-diseno-y-frameworks.html>.

Cuaresma, Sergio Blanco. 2008. Metodologías ágiles gestión de proyecto. *Metodologías ágiles Gestión de proyecto*. [En línea] 7 de junio de 2008. [Citado el: 7 de enero de 2012.] <http://www.marblestation.com>.

CUVIS. 2010. *XP_y_otras_Metodologias_Agiles*. 2010.

Cuba, postgresQL. 2012. postgresQL Cuba. *comunidad técnica cubana*. [En línea] 2012. [Citado el: 15 de febrero de 2012.] <http://postgresql.uci.cu/>.

definición.de. 2008. definición.de. *definición.de*. [En línea] 2008. [Citado el: 20 de noviembre de 2011.] <http://definición.de/cartografía/>.

Elizabeth De la Cruz Rodríguez, Dalianny Guzmán Hernández. 2010. *Propuesta de herramienta CASE para los proyectos del Centro de Desarrollo de Informática Industrial*. Habana: s.n., 2010. Tesis.

farlex, The free dictionary by. 2010. The free dictionary by farlex. [En línea] 2010. [Citado el: 27 de noviembre de 2011.] <http://es.thefreedictionary.com>.

Fernández., Miguel Romero. 2010. Proceso de Georeferenciación de la Cartografía Histórica. *Proceso de Georeferenciación de la Cartografía Histórica*. [En línea] Proceso de Georeferenciación de la Cartografía Histórica., 2010. [Citado el: 6 de diciembre de 2012.] <http://www.expobus.us.es/cartografía/salas/sala12/georeferenciación.html>.

Fabien Potencier, François Zaninotto. librosweb. librosweb. [En línea] [Citado el: 15 de Febrero de 2012.] <http://www.librosweb.es>

The free dictionary. 2010. The free dictionary by farlex. [En línea] 2010. [Citado el: 27 de noviembre de 2011.] <http://es.thefreedictionary.com>.

Bibliografía



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

Jiménez, Darwin y Aguirre, Carlos Eduardo. 2008. *Modelo de pruebas de software.* Slideshare. [En línea] 2008. [Citado el: 3 de Abril de 2012.] <http://www.slideshare.net/dajigar/presentación-pruebas-presentation>.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El proceso unificado de desarrollo de software.* Madrid : Addison Wesley, 2000.

Latina, osmosis. 2012. osmosis latina. *osmosis latina.* [En línea] 18 de febrero de 2012. <http://www.osmosislatina.com/lenguajes/uml/básico.htm>.



Mapserver. mapserver.org. *mapserver.org.* [En línea] <http://mapserver.org/es/index.html>.

NetBeans. NetBeans. [En línea] 2010. <http://netbeans.org/features/index.html>.

OpenLayers. 2012. OpenLayers. *OpenLayers.* [Online] OpenLayers, 2012. <http://openlayers.org/>.

Pressman, Roger S. 2005. *Ingeniería del Software un enfoque práctico.* La Habana: Félix Varela, 2005. págs. 159-187.

ANEXOS

Anexo 1. Entrevista

Entrevista realizada al líder del equipo de integración y componentes del proyecto Aplicativos SIG: Alain León Companioni.

1. ¿Cuáles son las funcionalidades que debe tener el Módulo de Georeferenciación Eventual?

Funcionalidades:

La primera funcionalidad debe permitir adicionar y eliminar un evento que será mostrado en el componente control de capas.

La segunda funcionalidad permitirá dibujar en el mapa líneas, puntos y polígonos los cuales contendrán información asociada que será visualizada por los usuarios en el momento que lo deseen.

Las geometrías adicionadas también deberán ser modificadas para ajustar su posición en el mapa.

Anexos

Anexo 2. Descripción de los casos de uso del sistema


Objetivo	Gestionar evento	
Actores	Usuario (Inicia).	
Resumen	El caso de uso se inicia cuando el usuario necesita añadir una capa para representar un determinado evento, para ello se le entra al sistema el nombre y la descripción del evento. Luego de ser añadido a la base de datos se mostrara en el componente control de capas todos los eventos listados y terminará el caso de uso.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Ninguna.	
Postcondiciones		
Flujo de eventos		
Flujo básico: Gestionar áreas.		
	Actor	Sistema
5.	Selecciona la opción "GeoreferenciarEventos" en la barra de menú.	
6.		Muestra la interfaz "Georeferenciareventos".
7.	Puede seleccionar las opciones para adicionar o eliminar un evento.	

Anexos

8.		<p>Habilita las opciones de Gestionar eventos.</p> <ul style="list-style-type: none"> • Adicionar evento. Ver Sección 1: “Adicionar evento”. • Eliminar evento. Ver Sección 2: “Eliminar evento”. <p>Termina el caso de uso.</p>
----	--	--

Sección 1: “Adicionar evento”

Flujo Básico: Adicionar evento.

	Actor	Sistema
5.	Selecciona la opción adicionar evento.	
6.		Muestra  campo para entrar en nombre del evento que será adicionado.
7.	Entra el nombre del evento y oprime el botón “RegistrarEvento”.	
8.		Se adiciona el evento a la base de datos y se muestra en el control de capas.



Sección 2: “Eliminar evento”

Flujo Básico: Eliminar evento.

Anexos

	Actor	Sistema
9.	Entra el nombre del evento que desea eliminar.	
10.		Se elimina el evento de la base de datos y se muestra un mensaje notificando la correcta eliminación.
Relaciones	CU Incluidos	Ninguno.
	CU Extendidos	Ninguno.
Requisitos no funcionales	Ninguno.	
Asuntos pendientes	No procede.	

Tabla 6 Descripción del CU Gestionar Evento

Objetivo	Adminis  imágenes.	
Actores	Usuario  (Inicia).	
Resumen	El caso de uso se inicia cuando el usuario necesita destacar un objetivo dentro de un área determinada y para ello selecciona una de las imágenes posibles a insertar.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Ninguna.	
Postcondiciones		
Flujo de eventos		
Flujo básico: Gestionar áreas.		
	Actor	Sistema

Anexos

9.	Selecciona la opción “GeoreferenciarEventos” en la barra de menú.	
10.		Muestra la interfaz “GeoreferenciarEventos”.
11.	Puede seleccionar las opciones para adicionar o eliminar una imagen.	
12.		<p>Habilita las opciones de Gestionar imágenes.</p> <ul style="list-style-type: none"> • Adicionar imagen. Ver Sección 1: “Adicionar imagen”. • Eliminar imagen. Ver Sección 2: “Eliminar imagen”. <p>Termina el caso de uso.</p>

Sección 1: “Adicionar imagen”

Flujo Básico: Adicionar imagen.

	Actor	Sistema
5.	Selecciona la opción adicionar área.	
6.		Muestra un combobox que se despliega con todas las posibles imágenes a pintar.

Anexos

7.	Selecciona una imagen y oprime el botón “Imagen”.	
8.		Se adicionan las imágenes al mapa dando clic sobre este.
Sección 2: “Eliminar imagen”		
Flujo Básico: Eliminar evento.		
	Actor	Sistema
9.	Entra el identificador de la imagen que se desea eliminar.	
10.		Se elimina la imagen de la base de datos y se muestra un mensaje notificando la correcta eliminación.
Relaciones		CU Incluidos
		Ninguno.
		CU Extendidos
		Ninguno.
Requisitos no funcionales		Ninguno.
Asuntos pendientes		No procede.

Tabla 7 Descripción del CU Administrar imágenes

Objetivo	Mostrar información.
Actores	Usuario :(Inicia).
Resumen	El caso de uso se inicia cuando el usuario necesita visualizar la información de algunas de las aéreas que se encuentran representadas sobre el mapa.

Anexos

Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Ninguna.	
Postcondiciones		
Flujo de eventos		
Flujo básico: Mostrar información.		
	Actor	Sistema
13.	Selecciona la opción “GeoreferenciarEventos” en la barra de menú.	
14.		Muestra la interfaz “GeoreferenciarEventos”.
15.	Seleccionar la opción para mostrar información.	
16.		Habilita la opción de mostrar información. Termina el caso de uso.
Sección 1: “Mostrar información.”		
Flujo Básico: Mostrar información.		
	Actor	Sistema
5.	Entra el identificador del área que se desea ver la información y se oprime el botón “Información”.	

Anexos

6.		Muestra un formulario con los datos del área seleccionada.
Relaciones	CU Incluidos	Ninguno.
	CU Extendidos	Ninguno.
Requisitos no funcionales	Ninguno.	
Asuntos pendientes	No procede.	

Tabla 8 Descripción del CU Mostrar información

Anexo 3: casos de pruebas

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo Central
SC1: Adicionar área.	EC 1.1: Crear área.	El usuario presiona un botón indicando el tipo de área a crear y comienza a dibujar	<ol style="list-style-type: none"> 1. Dar clic en uno de los botones(Punto, Línea o Polígono) 2. Se activa el control y se comienza a dibujar el área. 3. Clic en el botón “Detener” para desactivar el control de pintar.

Anexos

<p>SC2: Modificar un área.</p>	<p>EC 2.1: Modificar un área correctamente.</p>	<p>El usuario selecciona un tipo de área y la forma en que desea modificarla.</p>	<ol style="list-style-type: none"> 4. Se marca una de las opciones del Radiobutton. 5. Clic en el botón "Mover". 6. Clic sobre el área que será modificada. 7. Clic en el botón "Detener"
<p>SC3: Eliminar un área.</p>	<p>EC 3.1: Eliminar un área.</p>	<p>El usuario entra al sistema el id del área que desea eliminar</p>	<ol style="list-style-type: none"> 8. Se escribe el id del área en el textfile "IdArea". 9. Clic en el botón "EliminarAr". 10. Se muestra un mensaje notificando la correcta eliminación.

Tabla 9 Descripción del CP Gestionar áreas

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo Central
----------------------	-------------------------	---------------------------------	---------------

Anexos



SC1: Insertar imágenes.	EC 1.1: Insertar imagen.	El usuario despliega una lista de opciones de las posibles imágenes a insertar, selecciona una y pulsa un botón para comenzar la operación.	<ol style="list-style-type: none"> 1. Desplegar las opciones del combobox "Imágenes". 2. Se pulsa el botón "Imágenes". 3. Se activa el control y se pinta dando clic en el lugar donde se quiera insertar la imagen.
	EC1.2: Insertar imagen con fallo.	El usuario no despliega la lista de opciones de las posibles imágenes a insertar.	<ol style="list-style-type: none"> 4. Se pulsa el botón "Imágenes". 5. Se muestra un mensaje de error notificando el fallo.
SC3: Eliminar una imagen.	EC 3.1: Eliminar una imagen.		<ol style="list-style-type: none"> 6. 7. 8.

Tabla 10 Descripción del CP Administrar imágenes

Nombre de la sección	Escenario de la sección	Descripción de la funcionalidad	Flujo Central
----------------------	-------------------------	---------------------------------	---------------

Anexos

SC1: Mostrar información.	EC 1.1: Mostrar información.	El usuario entra al sistema el identificador de un área y este devuelve un formulario con la información referente a ella.	<ol style="list-style-type: none">1. Entrar el identificador en el texfile "Inf".2. Se pulsa el botón "Información".3. Se muestra un formulario con los datos del área.
	EC1.2: Mostrar información con fallo.	El usuario no entra al sistema el identificador del área.	<ol style="list-style-type: none">4. Se pulsa el botón "Información".5. Se muestra un mensaje de error notificando el fallo.

Tabla 11 Descripción del CP Mostrar información

Anexos

Anexo 4. Diagrama de clases del diseño

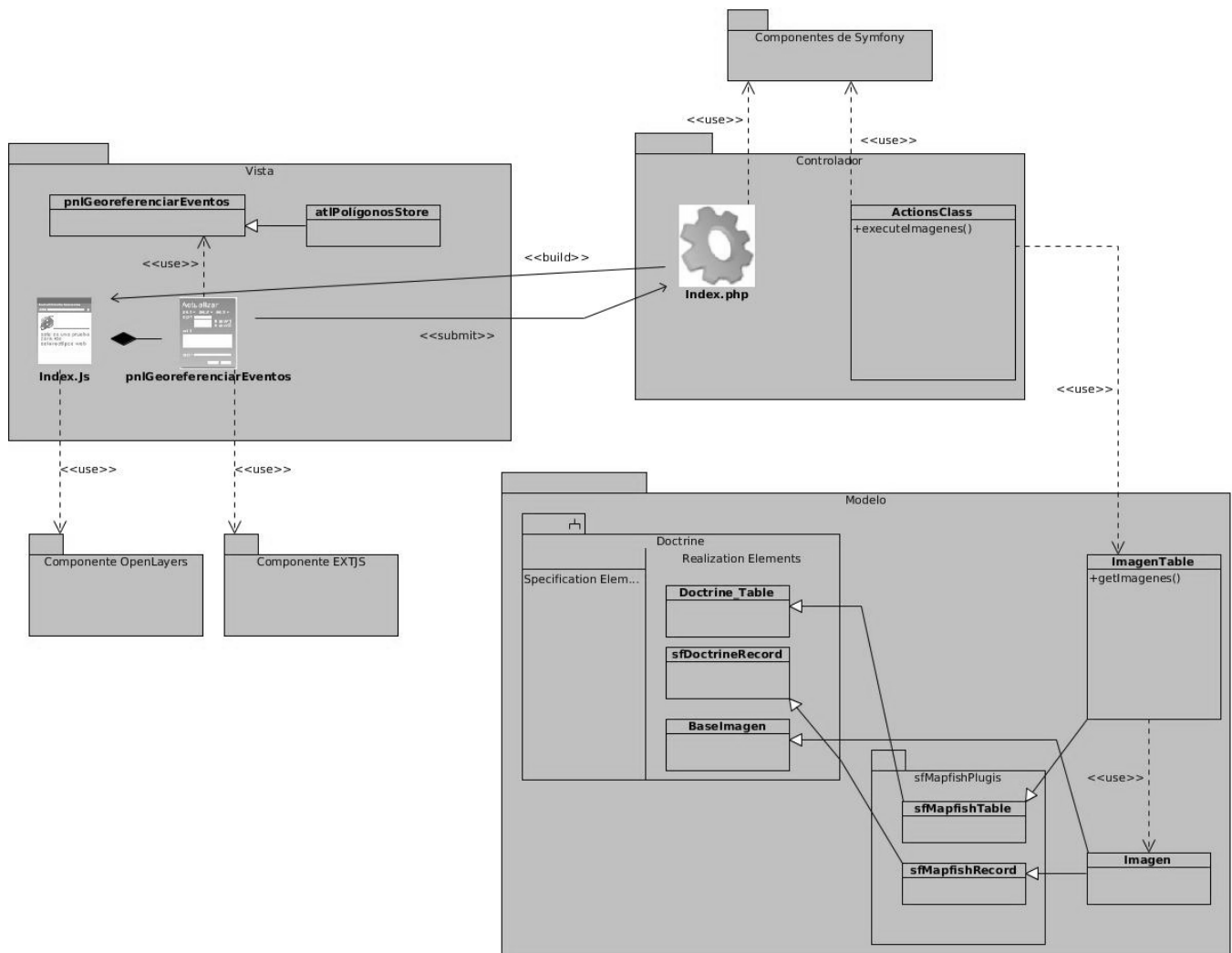


Imagen 10 Diagrama de clases del diseño del CU Administrar Imágenes

Anexos

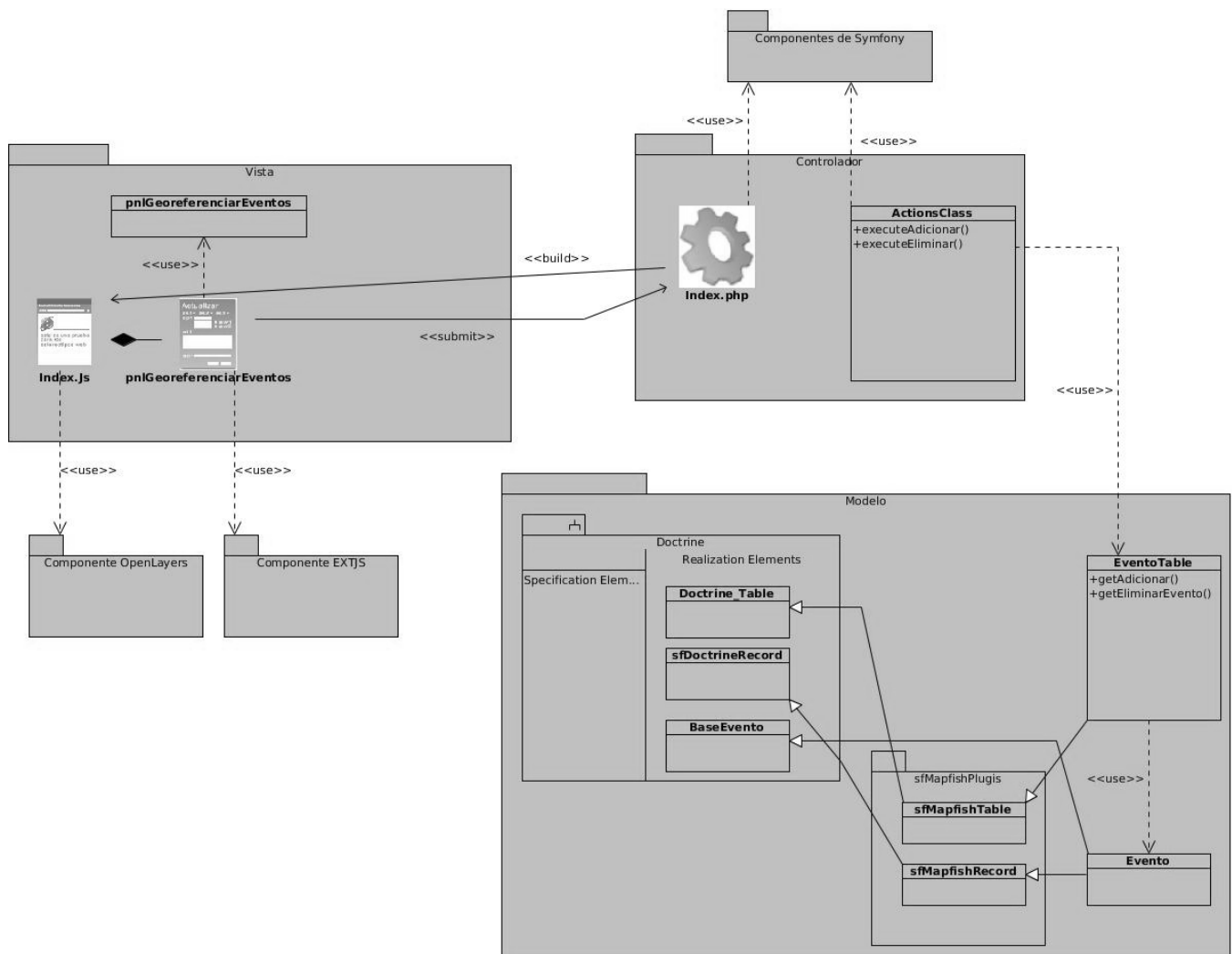


Imagen 11 Diagrama de clases del diseño del CU Gestionar eventos

Anexos

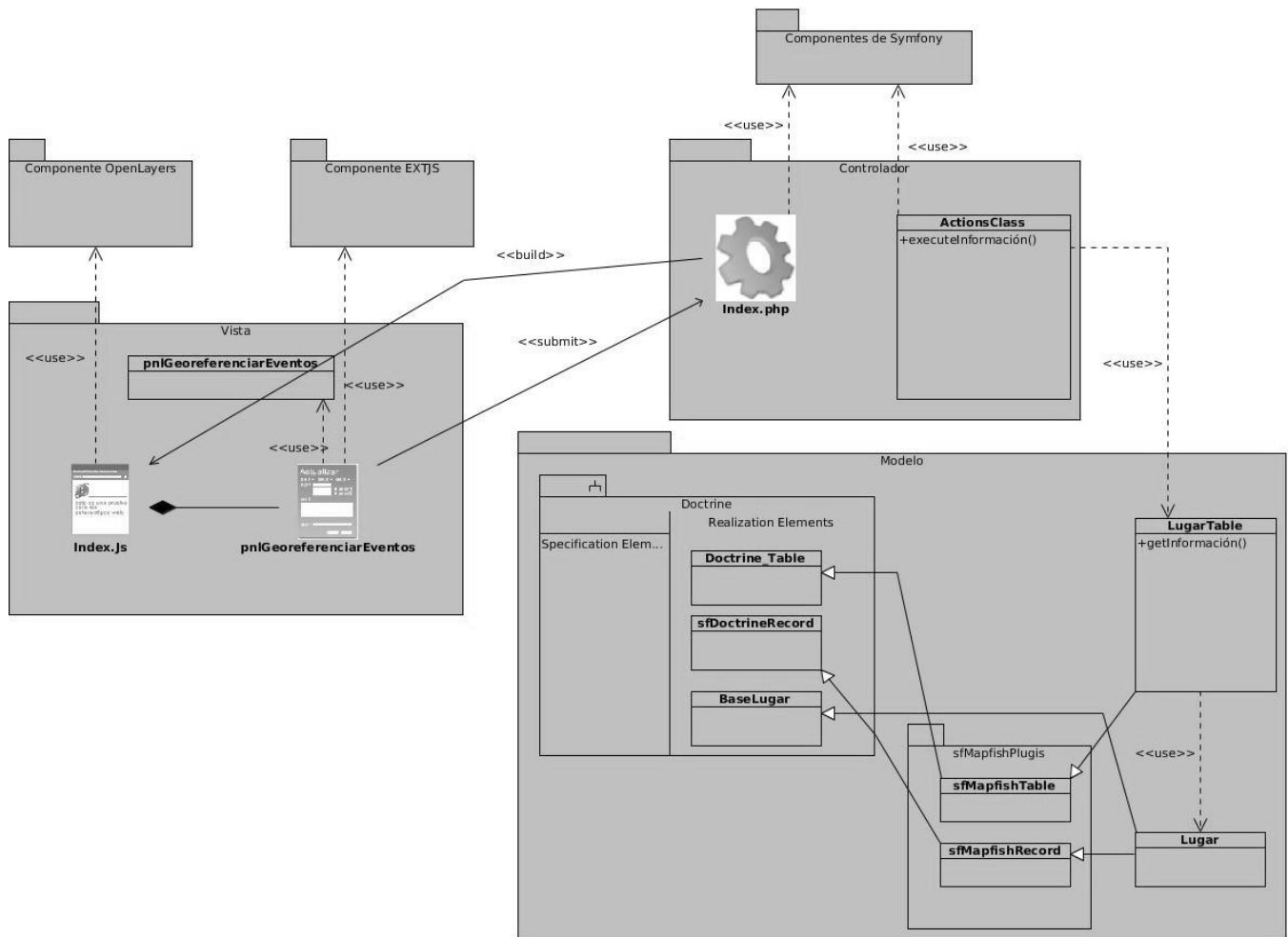


Imagen 12 Diagrama de clases del diseño del CU Mostrar Información

Glosario de Términos

GLOSARIO DE TÉRMINOS

API: (Application Program Interface). Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

Framework: Es un esquema, esqueleto o patrón para el desarrollo y/o la implementación de una aplicación.

HTML: Es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet.

Multiplataforma: Sistema informático que corre sobre varios sistemas operativos, sin prescindir de ninguna de sus funcionalidades.

Interfaces: Cara visible de los programas que interactúa con los usuarios.

Software: Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.