



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

Título: Plataforma para el desarrollo de Sistemas de Información Geográfica basados en GeoExt.

Autora:

➤ Lianet Maldonado Bosque.

Tutor:

➤ Ing. Pedro José Pérez González.

Universidad de las Ciencias Informáticas

Cuidad de la Habana, Junio 2011.

DECLARACIÓN DE AUTORÍA

Declaración de autoría

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los ____ días del mes de _____ del año 2011.

Lianet Maldonado Bosque

Pedro José Pérez González

Firma de la Autora

Firma del Tutor

Datos de Contacto

Síntesis de la Autora: Lianet Maldonado Bosque.

Correo Electrónico: lbosque@estudiantes.uci.cu.

Síntesis del Tutor: Ing. Pedro José Pérez González.

Profesión: Ingeniero en Ciencias Informáticas.

Años de graduado: Adiestrado.

Correo Electrónico: pjgonzalez@uci.cu.

Agradecimientos

A mi madre por haberme dado la vida, por ser mi eterna amiga, por su preocupación y comprensión, por su eterno amor y por serlo todo para mí.

A mi abue por ser la luz de mis ojos, mi ídolo, mi eterna inspiración, mi todo.

A mi herma por siempre estar ahí cuando lo necesito, por haber sido mi padre y hermano a la vez.

A mis tíos y mi hermana Oli por ser la mejor familia del mundo.

A mi amigo Alain Sánchez, por auxiliarme, quererme y apoyarme.

A mis amigas Sandra, Gelsis, Raisa, Thais, y Milayne por ser las hermanitas que no me dio la vida y sin embargo las tuve, por quererme tanto.

A mis amigos Alain López, Yusley, Pablo Luis, Aurelio, Alejandro y Yoandy por haberme ayudado y cuidado, demostrándome el valor de la amistad varonil.

A mi tutor Pedro, por haberme guiado, soportado y ayudado hasta lograr el título de Ingeniera en Informática.

A todas estas personas y muchas más que por un motivo u otro se me olvidó mencionar, pero pueden estar seguros que les agradezco.

A la vida gracias por haberme dado el premio de ser ingeniera...

Lianet Maldonado Bosque.

Dedicatoria

A mi madre, abuela, hermano, familia, amigos y a todas aquellas personas que me han acompañado durante esta etapa de mi vida y que de una u otra manera la enriquecieron personal y profesionalmente.

Muchas Gracias...

Resumen

Las Tecnologías de la Información y las Comunicaciones (TIC) han alcanzado un alto grado de importancia en el mundo moderno. Esto se debe a la necesidad de las personas de contar con un fácil acceso a todo tipo de información y a la necesidad de automatización de los procesos de negocio que se llevan a cabo en las empresas, como resultado del gran cúmulo de información que se maneja y con el objetivo de agilizar la gestión de dicha información.

La investigación se centra en el desarrollo de una plataforma que agilice el proceso de creación de Sistema de Información Geográfica (SIG) basados en GeoExt, en el Centro de Desarrollo de Geoinformática y Sistemas Digitales (GEySED), específicamente en el Departamento de Desarrollo de Software Geoinformática.

El desarrollo de la plataforma acelerará la creación de SIG, convirtiéndose en todo un escenario para futuros productos, objetivado en la rapidez y mejoramiento del entorno de trabajo.

Los resultados alcanzados a partir de la investigación realizada son recogidos a lo largo del documento, teniendo como objetivo realizar la implementación de las funcionalidades necesarias, definiendo las principales herramientas y librerías a usar.

Palabras claves:

- Geoinformática.
- Plataforma.
- Sistema de Información Geográfica (SIG).

Abstract

Information Technology and Communications (ICT) have achieved a high degree of importance in the modern world. This is due to the need for people to have easy access to all information and the need to automate business processes that take place in business as a result of the large amount of information is handled and with the aim of streamlining the management of such information.

The research focuses on developing a platform to accelerate the process of creating geographic information system (GIS) based on GeoExt, Development Center for Geoinformatics and Digital Systems (GEYSER), specifically in the Software Development Department Geo.

The development platform will speed the creation of GIS, becoming an entire stage for future products, objectified in the speed and improving the work environment.

The results obtained from the research are collected throughout the document, aiming to make the implementation of the necessary functions, defining the main tools and libraries to use.

Keywords:

- Geo.
- Platform.
- Geographic Information System (GIS).

Índice

Declaración de autoría	II
Datos de Contacto	III
Agradecimientos	IV
Dedicatoria	V
Resumen	VI
Abstract	VII
Introducción	1
Capítulo 1. Fundamentos Teóricos.	6
1.1 Introducción.	6
1.2 Conceptos asociados al dominio del problema.	6
1.2.1 Sistemas de Información Geográfica (SIG).	6
1.2.2 Plataforma.	7
1.3 Objeto de Estudio.	7
1.3.1 Descripción general y procesos comunes.	7
1.3.2 Descripción del domino del problema	8
1.4 Sistema de Información.	9
1.4.1 Sistema de Información Geográfica.	9
1.5 Soluciones existentes en el mundo y Cuba	11
1.5.1 Capaware	11
1.5.2 ArcView	11
1.5.3 GeneSig	11
1.6 Sistemas web	12
1.7 Plataforma de desarrollo	14

1.8 Conclusiones parciales -----	17
Capítulo 2. Herramientas y Tecnologías. -----	18
2.1 Introducción. -----	18
2.2 Metodología de desarrollo de Software. -----	18
2.2.1 Agile Unified Process (AUP) -----	19
2.2.2 Roles -----	20
2.3 Tecnologías web del lado del servidor -----	21
2.3.1 PHP -----	22
2.3.2 YAML -----	24
2.3.3 Estándar de codificación -----	25
2.3.3.1 Estándar de codificación para PHP -----	26
2.4 Tecnologías web del lado del cliente -----	27
2.4.1 ExtJs -----	28
2.4.2 OpenLayer -----	29
2.4.2.1 Herramientas disponibles -----	30
2.4.3 GeoExt -----	31
2.4.4 Estándar de codificación para Ext -----	31
2.5 Servidores -----	32
2.5.1 Apache -----	32
2.5.2 Postgres -----	33
2.5.3 Postgis -----	33
2.6 Herramientas -----	34
2.6.1 Visual Paradigm -----	34
2.6.2 Zend Studio -----	35
2.7 Conclusiones parciales -----	36
Capítulo 3. Análisis y Diseño. -----	37

3.1 Introducción.-----	37
3.2 Descripción del Modelo de Dominio -----	37
3.3 Especificación de los requisitos de la aplicación -----	38
3.3.1 Requisitos funcionales-----	38
3.3.2 Requisitos no funcionales-----	38
3.4 Definición de los actores y casos de usos del sistema-----	40
3.4.1 Actores del sistema-----	40
3.4.2 Diagrama de Casos de Uso del Sistema-----	40
3.4.3 Especificación de los Casos de Uso del Sistema -----	41
3.5 Estilo arquitectónico utilizado-----	49
3.6 Patrones de diseños utilizados-----	50
3.6.1 Patrón GRASP -----	50
3.6.2 Patrón SINGLETON-----	50
3.6.3 Patrón GOF -----	50
3.7 Diagramas de clases del diseño-----	50
3.8 Diagramas de colaboración-----	50
3.9 Conclusiones parciales -----	51
Capítulo 4. Desarrollo de la solución y validación. -----	52
4.1 Introducción-----	52
4.2 Modelo de implementación -----	52
4.2.1 CU: Atender petición -----	52
4.2.2 CU: Crear controlador -----	53
4.2.3 CU: Manejar sesiones de usuarios -----	53
4.2.4 CU: Cache-----	53
4.2.5 CU: Response -----	53

4.2.6 CU_Interfaz: Acercar-----	53
4.3 Modelo de despliegue-----	54
4.4 Pruebas unitarias-----	55
4.4.1 Diseño de las pruebas que permitan validar la solución propuesta.-----	55
4.4.1.1 Realizar Navegación.-----	55
4.4.1.2 Localizar por Coordenadas.-----	58
4.4.1.3 Medir distancia. -----	60
4.5 Pruebas funcionales-----	61
4.6 Conclusiones parciales -----	62
Conclusiones Generales -----	63
Recomendaciones-----	64
Bibliografía -----	65
Glosario de Términos -----	68

Índice de figuras

FIGURA 1 MODELO CLIENTE – SERVIDOR.....	14
FIGURA 2 EL CICLO DE VIDA DEL AGILE UP.....	20
FIGURA 3 . MODELO DE DOMINIO.	37
FIGURA 4 DIAGRAMA DE CASO DE USO DEL SISTEMA.	41
FIGURA 5 DIAGRAMA MODELO VISTA CONTROLADOR.	49
FIGURA 6 CU: ATENDER PETICIÓN.....	53
FIGURA 7 CU: ACERCAR.....	54
FIGURA 8 DIAGRAMA DE DESPLIEGUE DEL SISTEMA.	54

Índice de tablas

TABLA 1 ROLES DE AGILE UP.	21
TABLA 2 DESCRIPCIÓN TOTAL DEL ACTOR DEL SISTEMA.	40
TABLA 3 CUS REALIZAR NAVEGACIÓN.	45
TABLA 4 CUS LOCALIZADOR POR COORDENADAS.	46
TABLA 5 CUS MEDIR DISTANCIA.	49
TABLA 6 SECCIONES A PROBAR EN EL CASO DE USO REALIZAR NAVEGACIÓN.	56
TABLA 7 ACERCAR.	56
TABLA 8 ALEJAR.	57
TABLA 9 RECENTRAR MAPA.	57
TABLA 10 MOVER MAPA.	58
TABLA 11 SECCIONES A PROBAR EN EL CASO DE USO LOCALIZAR POR COORDENADAS.	58
TABLA 12 VARIABLES A PROBAR.	59
TABLA 13 LOCALIZAR POR COORDENADAS COMPLETO.	59
TABLA 14 SECCIONES A PROBAR EN EL CASO DE USO MEDIR DISTANCIA.	60
TABLA 15 MEDIR DISTANCIA.	60
TABLA 16 CALCULAR ÁREA.	61

Introducción

El mundo se encuentra en una revolución informática, la demanda de los productos de software y los servicios de información tecnológica tienen una de las tasas de crecimiento mundiales más alta en la actualidad. Las Tecnologías de la Información y Comunicación (TIC) juegan un papel relevante en la economía mundial. Su capacidad de atracción, de inversión y generación de valor se ve reflejada en la detonación de nuevas capacidades productivas, así como en la generación de empleos bien remunerados en diferentes economías del mundo. La industria del software y servicios informáticos (SSI) ha sido una de las más dinámicas a escala global en los últimos años. Esto no es sorprendente, considerando que el software juega un papel clave dentro del conjunto de avances tecnológicos.

La industria del software ha logrado significativos avances en diferentes facetas, como en la Economía, la Socialización del Conocimiento y en las Herramientas para manejar la información, por lo que se precisa que los productos que ella genera cuenten con la calidad requerida. Actualmente, las compañías de todo el mundo industrializado, reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general.

La desarrolladora de software no es la excepción y ha realizado intensos trabajos al respecto, estableciendo una estrecha relación entre software y calidad. Se hace énfasis en la calidad del software que va hacer la concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario. Calidad va implicar eficiencia, flexibilidad, corrección, confiabilidad, usabilidad, seguridad e integridad.

En los últimos decenios el tratamiento geográfico de la información ha cobrado un auge vertiginoso a escala mundial, cada día con mayores posibilidades de aplicación gracias al desarrollo de la aparición de los Sistemas de Información Geográfica (SIG).

Las expectativas creadas sobre SIG están también presentes en Cuba con sus correspondientes limitaciones y paradojas. Los usuarios de los SIG en Cuba no son los cibernéticos o especialistas informáticos, la mayoría son geólogos, cartógrafos, geógrafos, desarrolladores, arquitectos, informáticos, e ingenieros, quienes conocen y

operan SIG en sus investigaciones y proyectos. Muchos de estos especialistas han aprendido a manejar los SIG de forma autodidacta, sin un adiestramiento previo y trabajan en organizaciones oficiales del gobierno, tales como GeoCuba, Ministerio de Ciencia, Tecnología y Medio Ambiente, Instituto de Planificación Física, Ministerio de las Fuerzas Armadas y Universidades.

La mayoría de las universidades importantes a nivel mundial cuentan con Sistemas de Información Geográfica ya que se considera que entre el 80 y el 90 por ciento de la toma de decisiones tiene involucrada un componente geoespacial y no solo en las universidades sino de empresas, instituciones y el resto de las esferas de la sociedad.

Por otra parte, en los últimos tiempos se ha visto una inclinación a la combinación de educación en tecnologías de información espacial con otras carreras como las de Informática, lo cual constituye un antecedente importante para el diseño del Perfil de Geoinformática en la universidad surgida en la batalla de ideas.

La Universidad de las Ciencias Informáticas (UCI) surge con el firme propósito de impulsar el desarrollo de la informática y de la Industria de Software en Cuba. La docencia, la producción de software y la investigación son pilares inseparables que tributan a la formación de profesionales durante los cinco años de estudio de la carrera. Gracias a este vínculo y al trabajo que producen algunos profesores e informáticos en la universidad, se han creado disímiles portales y sitios web, productos Multimedia, Software de Gestión y otras aplicaciones, manteniendo convenios con organismos y entidades tanto nacionales como internacionales.

La universidad cuenta ya con prestigio a nivel nacional e internacional por su nivel de excelencia e informatización, donde su principal objetivo es la automatización de los procesos involucrados en el quehacer cotidiano, se encuentra inmersa en el desarrollo de software, convirtiéndose en la mayor productora del mismo en el país. Compuesta por 7 facultades, cada una con Centros de desarrollo de software. Específicamente la facultad 6 no se encuentra ajena y cuenta con dos centros, de datos (DATEC) y Geoinformática y Señales Digitales (GEySED).

En el Departamento de Desarrollo de Software Geoinformática, se trabaja en la realización de Sistemas de Información Geográfica (SIG). Desarrolla sobre plataformas ya que es la base para futuros productos, logrando agilizar el proceso de

construcción de SIG, alcanzando una mejora del tiempo empleado para el desarrollo de los sistemas anteriormente mencionados. Cuenta con una plataforma nombrada GeneSIG, la misma un proyecto integrado con la universidad, las Fuerzas Armadas Revolucionarias y GeoCuba, por lo que la toma de decisiones es compleja.

La realización de una nueva plataforma propicia un intercambio del cliente con su producto, personalizando cada una sus funcionalidades, mejorando considerablemente el proceso para el desarrollo rápido del Departamento de Geoinformática en la realización de SIG y permitiendo que se realice de manera segura y con todas las funcionalidades requeridas.

De lo planteado se identifica el **problema a resolver**:

¿Cómo agilizar la respuesta del Departamento de Geoinformática de la Universidad de las Ciencias Informáticas a clientes interesados en soluciones SIG para la web sobre la herramienta libre GeoExt?

Precisando como **objeto de estudio** de la investigación el proceso de desarrollo de Sistemas de Información Geográfica, enmarcado en el **campo de acción**: Proceso de desarrollo de la Plataforma Atlas.

De lo analizado se declara como **objetivo general** el desarrollar una plataforma para el desarrollo de Sistemas de Información Geográfica basados en GeoExt.

Como **idea a defender** se asume que con el desarrollo de la plataforma se agilizará el proceso de creación de Sistema de Información Geográfica basados en GeoExt.

Para el desarrollo de la investigación se trazan las siguientes **tareas investigativas** a cumplir:

- Caracterizar los procesos de desarrollo de los Sistemas de Información Geográfica.
- Seleccionar y argumentar la Metodología de Desarrollo de Software a usar en el proceso.
- Identificar las principales funcionalidades del sistema.
- Analizar y diseñar el sistema.
- Implementar el sistema.
- Validar el resultado obtenido.

Los **métodos científicos a utilizar** en esta investigación son:

Métodos Teóricos

Histórico lógico: Se conocerán las tendencias actuales en cuanto a la gestión de la información de los procesos, y uso de tecnologías de desarrollo.

Análisis y Síntesis: Se analizarán los documentos e información obtenida relacionada con el desarrollo de una plataforma de desarrollo para SIG en el Departamento de Geoinformática.

Métodos Empíricos

Observación: Se indagará en la ocurrencia de procesos con vistas a clasificar aspectos importantes del trabajo. Se utiliza inicialmente para el estudio del estado del arte, a la hora de elaborar la situación problemática, cuando se detectaron las consecuencias que trae para el estudiante el no tener un sistema de búsqueda que facilite la obtención de información. Aclarar que este método se utilizará a lo largo de toda la investigación.

Entrevistas: Serán realizadas a los participantes en los procesos donde a través de conversaciones se identifican elementos importantes del negocio.

Luego de la puesta en marcha de la plataforma resultante, y dando cumplimiento a todos los objetivos planteados se logrará mejorar considerablemente el proceso para el desarrollo rápido del Departamento de Geoinformática permitiendo que este se realice de manera segura y con todas las funcionalidades esperadas, disminuyendo el esfuerzo y tiempo de desarrollo de las tareas, obteniéndose como **resultados** la documentación general referente al proceso de desarrollo y la plataforma ligera para los SIG.

El presente trabajo de diploma se estructura de la siguiente manera:

Capítulo 1. Fundamentación Teórica: Se describe la situación actual en el Departamento de Geoinformática correspondiente a la Facultad 6 del desarrollo de una Plataforma ligera para los Sistemas de Información Geográfica, se exponen los principales problemas que motivan el desarrollo de esta investigación, describiéndose en qué consiste la propuesta y se hace referencia a otros sistemas automatizados existentes.

Capítulo 2. Herramientas y Tecnologías actuales a desarrollar: Se realiza un estudio referente a las herramientas y tecnologías existentes en la actualidad que han sido seleccionadas para la confección del proyecto, planteándose lo fundamentado en cada caso. Se definen los estándares de código a usar.

Capítulo 3. Análisis y Diseño de la solución propuesta: Se describen los procesos actuales a través de un modelo de dominio, el cual sirve de base para determinar qué se va a desarrollar. Se determinan las funcionalidades del sistema propuesto y se describen en detalle.

Capítulo 4. Desarrollo de la solución y validación: Se abordan aspectos relacionados con la construcción de la solución propuesta, reflejando los principales componentes y diagramas usados, así como una descripción de las principales pruebas realizadas al sistema.

Capítulo 1. Fundamentos Teóricos.

1.1 Introducción.

En el presente capítulo se brinda una descripción general y los conceptos asociados al dominio del problema que conciernen a esta investigación. Se identifican a través de la situación problemática los principales factores que impiden el óptimo funcionamiento de los procesos. Por último se analizan algunas soluciones existentes de gestión de información tanto a nivel nacional como internacional.

1.2 Conceptos asociados al dominio del problema.

El dominio del problema tiene asociado una serie de conceptos como SIG y plataforma, los cuales serán abordados a continuación.

1.2.1 Sistemas de Información Geográfica (SIG).

Los SIG son una integración organizada de hardware, software y datos geográficos, diseñados para capturar, almacenar, manipular, analizar y desplegar en todas sus formas, la información geográficamente referenciada con el fin de solucionar dificultades complejas relacionadas con los procesos de planificación y gestión. Igualmente, se definen por los especialistas como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres, construido para satisfacer necesidades concretas de información.

En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común con 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. Las Tecnologías de la Información han sido conceptualizadas como la integración y convergencia de la computación. (Sendra, 1992).

1.2.2 Plataforma.

La plataforma de desarrollo será el entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo; sin embargo, aquí se encontrará ligada a un grupo de lenguajes de programación o a una interfaz de programación de aplicaciones.

Es precisamente el principio, en el cual se constituye un hardware, sobre el cual un software puede ejecutarse/desarrollarse. Define un estándar alrededor el cual un sistema puede ser desarrollado. Una vez que la plataforma ha sido definida, se produce el software y el hardware apropiado para su uso. (Plataforma Informática, 2005).

Se entiende por plataforma de desarrollo de SIG un sistema que sirve como base para hacer funcionar y mejorar SIG, por lograr productos completos, sencillos y fáciles de entender.

1.3 Objeto de Estudio.

1.3.1 Descripción general y procesos comunes.

Se especifican los principales servicios que brindan los SIG, explicando en que consiste cada una de ellas.

En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones, demostrado en sus funcionalidades.

Los principales procesos con que cuentan los SIG son:

- **Realizar Zoom:** Requerimiento mediante el cual los usuarios puedan navegar por el mapa a partir de la ejecución de las diferentes variantes de zoom (acercar, alejar).
- **Recentar Mapa:** Coloca al mapa en su máxima escala, no importando el tamaño de la imagen.

- **Paneo:** Se puede mover el mapa variando con el puntero del ratón la posición de la vista que se presenta.
- **Localizar por Coordenadas:** Se obtiene información y ubicación geográfica de una de las estructuras del mapa, según el lugar que se desee localizar mediante las coordenadas de longitud y latitud. Se localiza cualquier objeto del mapa de acuerdo a las coordenadas (x, y) entradas en los campos de Longitud y Latitud respectivamente.
- **Medir Distancia:** Se calcula la distancia entre dos o más puntos a partir del trazado de los mismos con el ratón y visualizar el valor de la distancia entre los dos últimos vértices dibujados y la distancia acumulada (del primer al último vértice) en un panel lateral.
- **Calcular Área de una Región:** Se calcula el área de una región a partir del trazado de la misma con el ratón sobre el mapa, se visualizan los valores del área en un panel lateral.

1.3.2 Descripción del domino del problema

En el Departamento de Desarrollo de Software Geoinformática correspondiente a la Facultad 6, se trabaja sobre una plataforma nombrada GeneSIG sirviendo como soporte al desarrollo de aplicaciones de SIG en entornos web con tecnologías libres, cumpliendo además con las especificaciones OpenGIS, pero la misma es demasiado robusta, lo que trae consigo una demora en el inicio de la elaboración de los sistemas geográficos. No deja de ser muy completa e utilizable para aplicaciones de gran envergadura.

Se desea contar con una plataforma sencilla, ligera y rápida, para que el usuario de no querer utilizar la plataforma GeneSIG cuente con la plataforma Atlas, no repitiendo las clases y métodos comunes, y existiendo un estándar de desarrollo.

De no utilizar una de las dos plataformas, el usuario se ve obligado a empezar desde cero, ejemplo de esto es que de aspirar una nueva funcionalidad para su sistema se tiene que iniciar nuevamente el proceso de construcción de botones y organización de los mismos, por lo que se vuelve cargante y arcaico.

Deseando eliminar el proceso de creación de SIG sencillos con aplicaciones robustas se realiza la plataforma Atlas para el desarrollo de SIG en GeoExt, donde el proceso se vuelve dinámico y se personalicen todas las funcionalidades.

1.4 Sistema de Información.

Los SI¹ y las Tecnologías de Información (TI²) han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones, lo más importante, su implantación logra ventajas competitivas o reducir la ventaja de los rivales.

1.4.1 Sistema de Información Geográfica.

Las tecnologías de los SIG, se están empleando en el mundo en casi todas las ramas de la ciencia y la tecnología. Con el aumento diario del conocimiento se reafirma el criterio, resultando indispensable que los especialistas estén informados a partir de fuentes documentales confiables acerca de las tecnologías de punta que surgen o la habilidad de adaptarse rápidamente a entender, evaluar y hacer uso de las innovaciones que continuamente surgen en materia de tecnología de la información. Se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente con el objetivo de capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. (Alvaro de J. Carmona, 2009).

Características principales de los SIG.

Los SIG realizan un gran número de manipulaciones, sobresaliendo las superposiciones de mapas en corto tiempo, transformaciones de escala, la representación gráfica y la gestión de bases de datos, así como su administración y

¹**SI:** Sistema de Información es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad.

²**TI:** Aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

mantenimiento. Dentro de las características que se encuentran referente a ellos podemos resaltar: (Alvaro de J. Carmona, 2009).

- Consultan rápidamente las bases de datos, tanto espacial como alfanumérica, almacenadas en el sistema, con información exacta, actualizada y centralizada.
- Realizan pruebas analíticas complejas, rápidas y repitiendo modelos conceptuales en despliegue espacial, sin la necesidad de repetir actividades redundantes o tediosas.
- Minimizan los costos de operación e incremento de la productividad.
- Ayudan en la toma de decisiones con el fin de resaltar esfuerzos y realizar inversiones más efectivas.
- Comparan eficazmente los datos espaciales a través del tiempo (análisis temporal).
- Efectúan algunos análisis de forma rápida, que hechos manualmente resultarían largos y molestos.

Los componentes de los SIG.

Los programas de SIG proveen las funciones y las herramientas necesarias para almacenar, analizar y desplegar la información geográfica. Los principales componentes de los programas son: (Alvaro de J. Carmona, 2009).

- Herramientas para la entrada y manipulación de la información geográfica. Un sistema de manejador de base de datos (DBMS³).
- Herramientas que permitan búsquedas geográficas, análisis y visualización.
- Interfaz gráfica para el usuario (GUI) para acceder fácilmente a las herramientas.

Probablemente la parte más importante de un sistema de información geográfico son sus datos. Los datos geográficos y tabulares pueden ser adquiridos por quien implementa el sistema de información, así como por terceros que ya los tienen disponibles. Los SIG integran los datos espaciales con otros recursos de datos y

³**DBMS:** Sistema de manejador de base de datos que es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

puede incluso utilizar los manejadores de base de datos más comunes para manipular la información geográfica.

1.5 Soluciones existentes en el mundo y Cuba

El panorama de las plataformas geoespaciales ha crecido enormemente en estos últimos años, demostrado en la cantidad de productos que inician su desarrollo, todos en busca de la elección de una plataforma de desarrollo potente, flexible y dinámica. Se enuncian diferentes aplicaciones, con el objetivo de sintetizar características similares que puedan ser utilizadas en el desarrollo del sistema que se propone.

1.5.1 Capaware

Una plataforma de software libre para el desarrollo de aplicaciones geográficas 3D multicapa, que surge a partir de la iniciativa del Instituto Tecnológico de Canarias en colaboración con la Universidad de Las Palmas de Gran Canaria. Visualiza numerosas hectáreas, aquellas partes que no son visibles no se procesan, por lo que las cargas al procesador son las mínimas necesarias en todo momento y admite conectarse a servidores externos con protocolos OGC⁴ para obtener datos para su propio uso. Así mismo se puede configurar y administrar las capas de recursos y elementos que se pueden representar sobre el terreno. (BOCHER, 2009).

1.5.2 ArcView

Plataforma de desarrollo de SIG más usada en el mundo. Es una herramienta desarrollada por la empresa estadounidense ESRI, con ella se pueden representar datos georeferenciados, analizar las características y patrones de distribución de esos datos y generar informes con los resultados de dichos análisis. (Minas, 2009).

1.5.3 GeneSig

La plataforma constituye una herramienta informática ideada por un grupo de proyecto conformado por especialistas de GEOCUBA, las Fuerzas Armadas y la Universidad de las Ciencias Informáticas (UCI). Surge por la necesidad de contar con un producto

⁴**OGC:** Creado en 1994 y agrupa a 372 organizaciones públicas y privadas. Su fin es la definición de estándares abiertos e interoperables dentro de los SIG y de la World Wide Web.

soberano que sirva como soporte al desarrollo de aplicaciones de SIG en entornos web con tecnologías libres, cumpliendo además con las especificaciones OpenGIS que establece el Open Geospatial Consortium (OGC) que garantizan la interoperabilidad global entre los SIG y en consecuencia con la política de migración a software libre y de soberanía tecnológica que impulsa Cuba. (Zaldívar, 2009) .

1.6 Sistemas web

Se entiende por sistema web o aplicación web, a un conjunto de páginas web (estáticas o dinámicas), entiéndase por página web estática a aquella en que es mostrada siempre al cliente de la misma forma sin importar las veces que sea llamada y por página web dinámica a la página cuyo contenido cambia en dependencia de los permisos de acceso que tenga el usuario que la solicita, de la cantidad de veces que sea llamada por el usuario. (Guglielmetti, 2005) .

Características de los Sistemas Web

Las características principales de una aplicación web son: (González, 2010).

- Se encuentra alojada en un Servidor Web.
- Son accesibles mediante el internet, usando un navegador web.
- La lógica del sistema se ejecuta en el servidor, mientras que el cliente solamente representa los datos.
- El acceso al sistema puede ser público o restringido.
- La actualización del sistema no afecta ni depende del cliente.
- Los sistemas son multiplataforma, ya que pueden ejecutarse en cualquier Sistema Operativo que tenga un servidor web o un navegador web.

Ventajas de los Sistemas Web

Los sistemas web ofrecen numerosas ventajas que pueden ser utilizadas al máximo para darle solución a los problemas existentes, los cuales son: (Sánchez, et al., 2007).

- Compatibilidad Multiplataforma: Los sistemas web funcionan de forma independiente a la plataforma que esté usando el cliente. Se cuenta con soporte para la mayoría de los sistemas operativos.
- Actualización: La actualización de los sistemas web se realiza sin la intervención del cliente, sin necesidad de interferir en su trabajo.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- **Fácil Instalación:** Para usar un sistema web no es necesario la descarga e instalación de este, el único requerimiento que piden estos sistemas es un navegador web y una conexión a internet.
- **Usuarios Concurrentes:** Los sistemas web brindan la posibilidad de tener varios usuarios conectados al sistema, sin que ninguno interfiera en el trabajo del otro.

Desventajas de los Sistemas Web.

- El personal de desarrollo de software con las competencias necesarias para realizar desarrollo web, todavía es relativamente caro.
- No es tan fácil que el personal de sistemas acostumbrado a programar en un entorno de DOS o visual, asimile rápidamente la programación web. Es necesario invertir un tiempo considerable en capacitación.
- Es más difícil la programación web que la programación en un entorno visual, ya que para manejar las interfaces se debe saber 4 lenguajes por lo menos. (Alfaro, 2007).

Arquitectura de los Sistemas Web

Los sistemas web en pocos años han evolucionado desde sencillos sistemas estáticos a complejos sistemas con una detallada lógica de negocio y una interfaz altamente configurable lo más parecida a las aplicaciones de escritorio. Debido a que en la actualidad, estos sistemas dan servicios a procesos de negocio de contable envergadura, estableciendo complejos requisitos de accesibilidad y tiempo de respuesta, ha sido necesario modificar y mejorar las técnicas de la arquitectura y diseño. (Garrido, 2004) .

Modelo Cliente – Servidor.

El uso de las aplicaciones que se encuentran publicadas en la Internet, es llamado modelo Cliente – Servidor. Cuando se hace uso de cualquier servicio que se encuentre en la Internet, ya sea visitar un sitio, consultar una Base de Datos, realizar una transferencia de ficheros; se establece un proceso en el cual intervienen dos integrantes. El cliente o usuario que ejecuta una aplicación en el ordenador local, el que se denomina programa cliente. Este cliente se encarga de comunicarse con el ordenador remoto para solicitar el servicio. El ordenador remoto atiende la solicitud del

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

cliente mediante un programa que se denomina programa servidor. Los términos Cliente – Servidor son usados para referirse a los programas que cumplen con estas características. (González, 2010) .

IBM define el modelo Cliente – Servidor como “...la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores”. (Alfaro, 2005) .

Los programas usados por los clientes para realizar las peticiones al servidor, realizan dos funciones principales; son los encargados de gestionar la comunicación con el servidor para solicitar el servicio deseado y recibir los datos enviados como respuesta, además son las herramientas que muestran los datos en la pantalla y ofrecen las funcionalidades necesarias para hacer uso de las prestaciones que ofrece el servidor.

El modelo Cliente – Servidor (véase Figura 1) facilita la integración y comunicación entre diferentes sistemas. De esta manera se puede usar sistemas que usen máquinas con diferentes sistemas operativos o sistemas medianos o grandes.

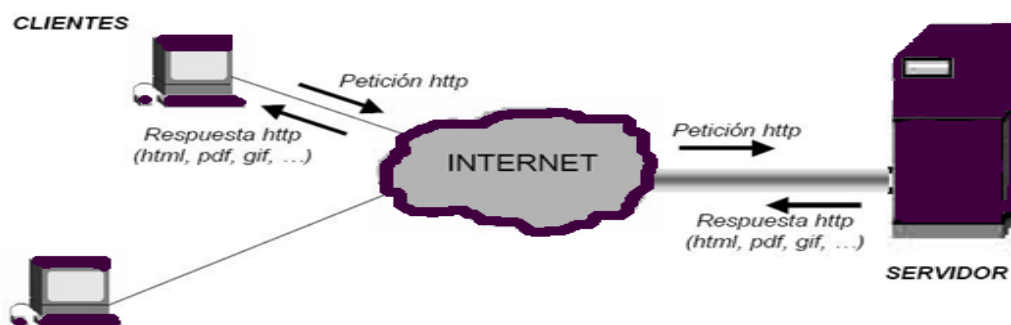


Figura 1 Modelo Cliente – Servidor.

1.7 Plataforma de desarrollo

Plataforma

Es el entorno de software común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Constituye un hardware, sobre el cual un software

puede ejecutarse/desarrollarse. Define un estándar alrededor el cual un sistema puede ser desarrollado, obteniéndose una plataforma de desarrollo de aplicaciones extremadamente potente, adaptable y confiable.

Características de la Plataforma para el Desarrollo de Aplicaciones SIG

Se ha desarrollado íntegramente una metodología para la creación de SIG en Internet, de fácil manejo y gran flexibilidad. A continuación se presenta la estructura de dicha metodología: (CODIMA, 2004).

Esta metodología incluye todo el Software necesario para poder integrar todos los elementos que constituyen un SIG.

Software:

Los programas SIG proveen de las herramientas y funcionalidades necesarias para almacenar, analizar y mostrar información geográfica. Los componentes principales del software SIG son:

- Sistema de manejo de base de datos: Incluye un potente asistente que permite la vinculación a la mayoría de los formatos Standard de Bases de Datos.
- Una interfaz gráfica de usuarios (IGU) para el fácil acceso a las herramientas: Su sencillez permite a usuarios no expertos interactuar fácilmente con todos los elementos.
- Herramientas para captura y manejo de información geográfica.
- Herramientas de soporte en consultas, análisis y visualización de datos geográficos: Incluye potentes asistentes para la generación y ejecución de consultas, así como visualización de resultados, gráficos, mapas temáticos, y callejeros.

Uso y Flexibilidad

Por sus características, puede utilizarse la misma plataforma para crear proyectos de índole tan distinta como necesidades tenga el usuario, entre los cuales se puede destacar los siguientes ámbitos de actuación: (CODIMA, 2004) .

- Social:

Interpretación de datos educativos, sanitarios y demográficos sobre un territorio.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

➤ Medio Ambiental:

Interpretación de datos ambientales como: Recursos Hídricos, Residuos Sólidos Urbanos y Gestión del Mobiliario Urbano.

➤ Económico:

Interpretación de datos empresariales: Ventas, Clientes y Productos.

Frameworks

Se utiliza este término para referirse a una estructura de software compuesta por una serie de componentes los cuales pueden ser personalizados dependiendo de las necesidades que presente la aplicación que se va a crear. Se puede decir que es una aplicación genérica configurable a las que se le pueden agregar piezas para construir un sistema completo. Para el desarrollo de la investigación se va a utilizar el framework Ext JS. El mismo es un API (Application Programming Interface.) escrito en Java Script con la finalidad de asistir el desarrollo de aplicaciones enriquecidas para internet (RIA⁵). Entre las principales características de Ext JS resalta un fuerte paradigma basado en componentes soportado por recursos para la programación orientada a objetos en Java Script que facilitan la implementación de extensiones y aplicaciones de gran complejidad, el poseer un amplio conjunto de widgets (pequeña aplicación o programa usualmente presentado en archivos o ficheros) configurables de alta calidad y una implementación transparente y sencilla para el trabajo con AJAX⁶. (Lobo, 2006).

Entre los principales objetivos de los frameworks de desarrollo se encuentra el de acelerar el proceso de desarrollo, reutilizar el código ya existente y promover las buenas prácticas de programación mediante el uso de patrones. Un framework de desarrollo web se define como un conjunto de componentes basado en un lenguaje web (PHP, Java, Python, Ruby, etc.), que componen un diseño reutilizable que facilita y agiliza el desarrollo web.

⁵**RIA:** Rich Internet Applications.

⁶**AJAX:** Acrónimo de Asynchronous JavaScript + XML(Introducción a AJAX, Javier Eguiluz Pérez).

En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta.

Características de los frameworks web

Características fundamentales con que cuentan los frameworks de desarrollo web, las que permiten el amplio uso de estos y su gran aceptación para la elaboración de sistemas web.

- **Abstracción de URL y Sesiones:** No es necesario manipular las URL ni las Sesiones, el framework lo hace de forma interna.
- **Acceso a datos:** Incluyen las herramientas y clases necesarias para interactuar con la mayoría de la Base de Datos (PostgreSQL, Oracle, MySQL, y MS SQL).
- **Control de peticiones:** Cuentan con clases controladoras que se encargan de atender las peticiones hechas por los clientes, estas son altamente adaptables.
- **Autenticación y Control de Acceso:** Tienen mecanismos de control de autenticación y control de acceso automático, los que se pueden adecuar a las necesidades de los clientes.

Si la medida para calificar un framework o una plataforma de desarrollo como buena o mala es el número de líneas de código que te toma hacerla, se cree que esa medida es bastante mala. La mejor plataforma será entonces una desarrollada específicamente para hacer un SIG determinado, donde se toma máximo una línea de código porque para eso fue hecho el framework. Pero se estaría ignorando por completo temas de escalabilidad, robustez y flexibilidad. (Montesino, 2009).

1.8 Conclusiones parciales

Luego de finalizar el capítulo se arribó a la conclusión de que es importante recalcar que ninguno de los sistemas mencionados cumple los requisitos necesarios para darle solución a los problemas existentes en el Centro de desarrollo de software GEYSED, debido a la inexistencia de una plataforma sencilla, manuable. El más adecuado a darle solución a las necesidades es GeneSIG, pero al no poder adecuarse a las nuevas necesidades del centro no es una posible solución.

Capítulo 2. Herramientas y Tecnologías.

2.1 Introducción.

Debido al surgimiento y utilización de Internet, se han abierto infinidad de posibilidades en cuanto al acceso a la información, representando un desafío a los desarrolladores, ya que los avances en tecnología demandan cada vez aplicaciones rápidas, y robustas que permitan utilizar y explotar las ventajas de la web. En este capítulo se realiza una descripción de las herramientas y tecnologías a utilizar en el desarrollo de un sistema web. Se aborda de forma general la metodología de desarrollo usada. Se definen los estándares de codificación usados en los leguajes especificados, tanto para el servidor como para el cliente.

2.2 Metodología de desarrollo de Software.

Todo proceso de desarrollo de una aplicación informática debe estar regido y orientado por una metodología de desarrollo de software, que guíe los procesos y permita tener un registro detallado del avance de la investigación. Las metodologías pueden ser ágiles o robustas, las metodologías ágiles están orientadas a la realización del software de una manera más objetiva, están generalmente enfocadas al desarrollo, no haciendo mucho énfasis en la documentación y su principal objetivo es garantizar la agilidad en el proceso de desarrollo.

Por su parte las metodologías robustas o pesadas están concebidas para guiar el proceso de desarrollo de los software de gran envergadura, cuando un proyecto requiere de gran cantidad de documentación por la complejidad de sus funcionalidades y este proyecto va a ser realizado en un tiempo considerablemente largo, además existe la posibilidad de que pase por las manos de varios equipos de trabajo, verdaderamente es necesario que exista una correcta documentación para que en un momento determinado los procesos de desarrollo puedan ser continuados por otras personas, entonces es que se debe utilizar una metodología robusta que garantice que todos los elementos estén en un orden perfectamente comprensible.

El Proceso Unificado Racional (Rational Unified Process, RUP) es una metodología de desarrollo de software usada en la ingeniería de software, con el objetivo de garantizar un desarrollo eficiente y robusto; proporciona una serie de diagramas y componentes usados en el ciclo de desarrollo de un sistema, obteniendo como resultado un proceso

dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. (Jacobson, 2000).

RUP es un proceso que define “quién” está haciendo, “qué” está haciendo, “cuándo” lo hace y “cómo” lo hace, utiliza para ello el Lenguaje de Modelado Unificado (Unified Modeling Language, UML), para crear los esquemas del sistema de software. (Jacobson, 2000).

RUP es un proceso estándar y flexible, al que se le pueden realizar variaciones en dependencia de la aplicación que se desea desarrollar. Esta metodología, además de usarse en el desarrollo de software también se ha usado en otras ramas y procesos industriales. (Jacobson, et al., 2004).

2.2.1 Agile Unified Process (AUP)

El Proceso Unificado Ágil (AUP) es una versión simplificada de RUP. Este describe un enfoque simple y fácil de entender para el desarrollo de software usando técnicas y conceptos que aún se mantienen vigentes en RUP. Se intentó mantener al Agile UP tan simple como fuera posible, ambos en este enfoque y en esta descripción. La descripción es simple. Los enfoques aplican técnicas ágiles incluidas en el Desarrollo Dirigido por Pruebas (TDD⁷), Desarrollo Dirigido por Modelado Ágil (AMDD⁸), administración de cambios ágil, y refactorización de bases de datos para mejorar la productividad. (Oviedo, 2006).

La Figura 1 representa el ciclo de vida de AUP. Lo primero que realiza es un aviso de que las disciplinas han cambiado. La disciplina de Modelado abarca las disciplinas de Modelado del Negocio, de Requerimientos y de Análisis y Diseño de RUP. El modelado es una parte importante en AUP, como se observa, pero no domina el proceso, puede seguir creando modelos y documentos los cuales son apenas lo suficientemente bueno. Segundo, las disciplinas de la Administración de la Configuración y Cambios ahora es la disciplina de la Administración de la

⁷**TDD:** Término usado para de notar el desarrollo dirigido por pruebas que es una práctica de programación que involucra otras dos prácticas: Escribir las pruebas primero y refactorización.

⁸**AMDD:** Desarrollo dirigido por modelado ágil enfocado en la metodología AUP.

CAPÍTULO 2. HERRAMIENTAS Y TECNOLOGÍAS

configuración. En el desarrollo ágil las actividades de administración de cambios son parte del esfuerzo de la administración de requerimientos, la cual es parte de la disciplina de Modelado.

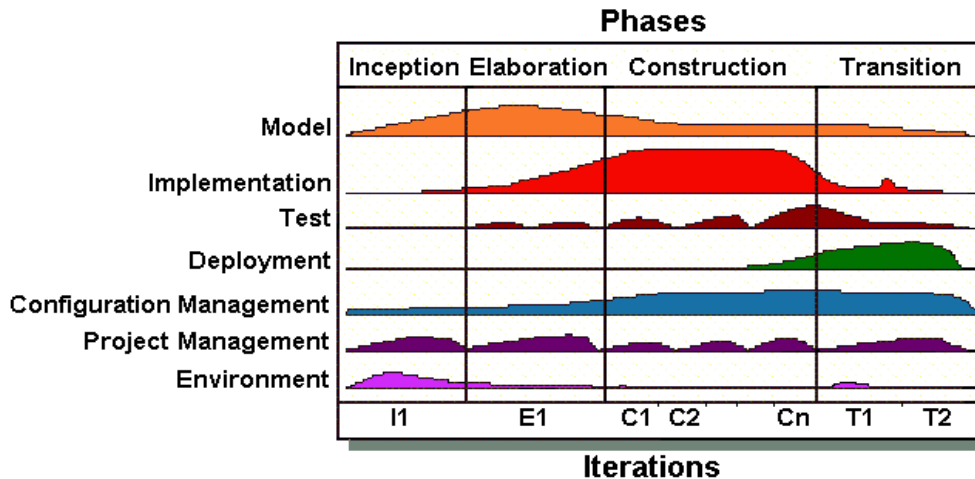


Figura 2 El ciclo de vida del Agile UP.

La naturaleza serial en Agile UP es capturada en cuatro fases: (Oviedo, 2006).

- Iniciación. El objetivo es identificar el alcance inicial del proyecto, una arquitectura potencial de su sistema, y obtener la financiación inicial del proyecto y la aceptación del involucrado.
- Elaboración. El objetivo es mejorar la arquitectura del sistema.
- Construcción. El objetivo es construir software funcional en una base regular e incremental, la cual cumpla con las necesidades de prioridad más alta de los involucrados de su proyecto.
- Transición. El objetivo es validar y desplegar el sistema en un ambiente de producción.

2.2.2 Roles

Los roles pueden ser asumidos por varias personas. En la Tabla 1 se describen brevemente cada uno de ellos, teniendo en cuenta que un rol no es un puesto. Se debe tratar de convertirse en un especialista general que domine una o más especialidades (por ejemplo, administración de base de datos, administración de proyectos, entre otras), un conocimiento general de todo el proceso del software y una gran comprensión del dominio de sus labores. (Oviedo, 2006).

CAPÍTULO 2. HERRAMIENTAS Y TECNOLOGÍAS

Rol	Descripción	Disciplina(s)
Administrador de la configuración	Encargado de proporcionar la infraestructura y crear el medio ambiente para el equipo de desarrollo.	Administración de la configuración
Implementador	Responsable de proporcionar al sistema los ambientes de pre-producción y producción.	Desarrollo
Desarrollador	Encargado de escribir código, realizar pruebas y construir el software.	Modelado, Implementación y Desarrollo
Especialista del proceso	Responsable de desarrollar, adaptar y apoyar el material de los procesos de la organización (descripción de procesos, plantillas, guías, ejemplos).	Entorno
Administrador de pruebas	Responsable del éxito de las pruebas, incluye planificar la administración, y promover las pruebas y las actividades de calidad.	Prueba
Especialista en herramientas	Responsable de seleccionar, adquirir, configurar y brindar mantenimiento al equipo requerido.	Entorno

Tabla 1 Roles de Agile UP.

Para la plataforma de desarrollo de sistemas de información geográfica se tendrá en cuenta cada uno de los roles explicados, explotando al máximo los mismos, definiendo las actividades, las cuales los miembros del equipo ejecutan para construir, validar y liberar el software, cumpliendo con las necesidades de sus involucrados.

2.3 Tecnologías web del lado del servidor

Podemos hablar de lenguajes del lado servidor asumiendo que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Las tecnologías, herramientas y lenguajes que se reflejan a continuación son ejecutados y utilizados en el servidor web. A diferencia de las que se ejecutan en el lado del cliente, estas no dependen de un navegador web, sino que son interpretadas por el servidor.

Un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro, o respecto a las versiones del mismo. En estos se procesa las peticiones de los clientes, se encuentra implementada la lógica del sistema, se valida la seguridad del sistema y la integridad de los datos enviados por los usuarios.

2.3.1 PHP

PHP es el acrónimo de Preprocesador de Hipertexto (Hypertext Preprocessor). Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran biblioteca de funciones y variada documentación. Es un lenguaje Open Source⁹ interpretado de alto nivel, pensado para desarrollar sistemas web, el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, dándole la ventaja de la facilidad de aprendizaje. La meta de este lenguaje es permitir a los desarrolladores escribir páginas web dinámicas de manera fácil y rápida.

Con el auge de la Programación Orientada a Objeto (POO¹⁰), se creó en el 2002 la versión de PHP 5, está basada en las características fundamentales de la POO; se eliminó el uso de la programación estructurada, se mejoró el uso de las clases, la herencia y la redefinición de métodos. (Angel Alvarez, 2004).

La POO es una forma especial de programar, más cercana a cómo expresaríamos las cosas en la vida real que otros tipos de programación. Con ella hay que aprender a pensar las cosas de una manera distinta, para escribir los programas en términos de objetos, propiedades, métodos. (Angel Alvarez, 2004).

La POO no es difícil, pero es una manera especial de pensar, a veces subjetiva de quien la programa, de manera que la forma de hacer las cosas puede ser diferente según el programador. Aunque podamos hacer los programas de formas distintas, no

⁹**Open Source:** Término usado para denotar al Código Libre, la característica fundamental es que el código de una aplicación no puede ser privado, sino que todos tienen que tener acceso a ver, analizar o modificar este código.

¹⁰**POO:** La programación orientada a objetos es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

CAPÍTULO 2. HERRAMIENTAS Y TECNOLOGÍAS

todas ellas son correctas, lo difícil no es programar orientado a objetos sino programar bien. Programar bien es importante porque así se pueden aprovechar todas las ventajas de la POO. (Angel Alvarez, 2004).

Con el uso de la POO en PHP 5 se resolvieron algunos de los principales problemas que existían en las versiones anteriores, tales como la clonación de objetos, que se realizaba al asignar un objeto a otra variable o al pasar un objeto por parámetro en una función. Para solventar este problema PHP 5 hace uso de los manejadores e identificadores de objetos, que son una especie de punteros que apuntan hacia los espacios en memoria donde residen los objetos. Cuando se asigna un manejador de objetos o se pasa como parámetro en una función, se duplica el propio manejador de objeto y no el objeto en sí.

Características de PHP 5 y la POO. (Angel Alvarez, et al., 2007).

Nombre fijo para los constructores y destructores: Los nombres a usar en los constructores y destructores están definidos de forma implícita y no pueden ser modificados (`__construct()`, `__destruct()`). Estos métodos se encargan de realizar las tareas de inicialización y destrucción de los objetos.

Acceso public, private y protected a propiedades y métodos: A partir de esta versión hay que usar los modificadores de acceso habituales a la POO. Estos métodos sirven para definir qué variables y métodos son accesibles desde cada entorno.

Uso de interfaces: Las interfaces son usadas en la POO para definir un conjunto de métodos que implementa una clase. Una clase puede implementar un conjunto de interfaces. En el desarrollo de aplicaciones es muy usado para suplir la falta de herencia múltiple en lenguajes como PHP y Java.

Método y clases “final”: Se puede definir que una clase o un método es “final”. En caso de que un método sea definido como “final”, este método no puede ser redefinido mediante la herencia. En caso de que lo definido como “final” sea una clase, esta no puede ser heredada por otra clase.

Operador instanceof: Operador usado para saber si un objeto es instancia de una clase determinada.

Atributos y métodos static: Se pueden definir métodos y atributos static, estos son las propiedades y funcionalidades a las que se puede acceder mediante el nombre de la clase, sin necesidad de realizar una instancia de la clase.

Clase y métodos abstractos: Es posible crear clases y métodos abstractos. Las clases abstractas no se pueden instanciar, estas se usan para heredarlas desde otras clases que pueden o no ser abstractas. Los métodos abstractos no se pueden llamar, se utilizan para ser heredados por otras clases las que los redefinen según las necesidades propias de las clases.

Constantes de clases: Se pueden definir constantes dentro de la clase, luego estas pueden ser accedidas a través de la propia clase.

Funciones que especifican la clase que reciben por parámetro: Se puede definir funciones donde se especifique el tipo de objeto que se le pasa como parámetro. En caso de no ser de este tipo se produce un error.

Función __autoload: Es habitual que en el desarrollo de un sistema se escriba un archivo por cada clase que se define, como técnica para organizar el código de las aplicaciones, pero debido a esto a veces es muy tedioso realizar la inclusión de cada clase. La función __autoload(), sirve para incluir el código de una clase que todavía no haya sido incluida en el código que se está ejecutando.

Clonado de objetos: Se puede crear un objeto a través de la copia exacta de otro objeto, utilizando la instrucción "clone". También se puede definir el método __clone() para realizar tareas asociadas con la clonación de un objeto.

2.3.2 YAML

YAML no es un lenguaje de marcado (del inglés YAML Ain't Markup Language). Es un formato para serializar datos de forma tal que es fácil procesarlos por la máquina, fácil de leer por las personas y fácil interactuar con los lenguajes de script. YAML permite escribir los datos como en XML, pero con una sintaxis mucho más sencilla, es un formato especialmente útil para describir los datos que pueden ser transformados en arreglos simples y asociativos. (Potencier, y otros, 2009).

YAML es mucho más rápido de escribir que XML, ya que no hacen falta las etiquetas de cierre y el uso continuo de las comillas y es mucho más poderoso que los tradicionales archivos .ini, ya que estos últimos no soportan la herencia y las estructuras complejas. Por este motivo, Symfony¹¹ utiliza el formato YAML como el lenguaje preferido para almacenar su configuración. (Potencier, et al., 2009).

YAML está diseñado para ser usado de forma fácil y amigable por personas que trabajen con datos. Se usan caracteres Unicode¹² para definir la estructura de información del fichero y el otro valor representa el valor de la variable. Logran organizar de forma mínima la enorme cantidad de datos con que se trabaja en algunas aplicaciones web. Está siendo usado en las nuevas versiones de lenguajes de desarrollo rápido como PHP, Perl, Python, Ruby y JavaScript.

En el desarrollo de un sistema web es necesario saber varios lenguajes de programación, pero solo es necesario uno para almacenar y transferir datos. YAML fue creado para trabajar con ficheros específicos como: ficheros de configuración, archivo de registros, lenguaje cruzado de transferencia de datos, persistencia y depuración de objetos en complejas estructuras de datos. (Ben Kiki, et al., 2006).

2.3.3 Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, debe establecerse un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado

¹¹**Symfony:** Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características.

¹²**Caracteres Unicode:** Caracteres definidos en una tabla donde cada carácter tiene un valor numérico. Utilizados para lograr la estandarización entre varios sistemas.

anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente. (Microsoft, 2009).

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La estabilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores o mejorar el rendimiento. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas.

2.3.3.1 Estándar de codificación para PHP

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y, posteriormente, se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener, a continuación se describe como se desea el estándar para php:

- No usar al tab para separar el código, use 2 espacios en blanco.
- No usar los espacios después de abrir y antes de cerrar los paréntesis.
- Use la notación camello o camelCase para las variables, métodos y funciones, esta notación define que los nombres empiezan por minúscula, en caso de ser un nombre compuesto el primer nombre empieza con minúscula y el segundo seguido empieza con mayúscula.
- Las llaves {} siempre se definen en una línea, se deben usar para definir el cuerpo de las estructuras de control y ordenar las expresiones definidas en el cuerpo de estas.
- Cada método de la clase o definición de miembro debe tener declarado de forma explícita la visibilidad de este, usando las palabras reservadas public, private, protected.

- Los ficheros .php no deben terminar con la etiqueta de cierre `?>`, esto no es necesario y puede provocar errores de difícil detección, en caso de que exista un espacio después de esta etiqueta.
- En el cuerpo de una función la declaración de retorno debe de estar separada por una línea en blanco, de esta forma se aumenta la legibilidad del código.
- Todo comentario que solamente tenga una línea, tiene que ser escrito usando la etiqueta de comentario de línea seguido por un espacio y el comentario en minúscula.
- Se debe evitar evaluar variables en conjunto con un string, en lugar de eso se debe usar la concatenación.
- Se deben usar las variables constantes definidas en php de forma minúscula, tales como `true`, `false`, `null`, `array`. De forma contraria cada vez que se defina una constante propia se debe declarar en mayúscula, de la siguiente forma: se define `(‘PI’,3.14)`, o una clase explícitamente para las constantes.
- Para comprobar si una variable es nula no se debe usar la función nativa de `phpis_null`, sino el operador de comparación exactamente distinto `(!==)`.
- Cuando se compara una variable con un string, se escribe primero el string y después la variable, en caso de poderse usar el operador de exactamente igual `(===)`.
- Se debe usar el nombre del tipo de dato en la declaración de los parámetros de los métodos y las funciones.
- Cada función y cada método declarado deben contar con la documentación php, donde se explique de forma clara y sencilla el funcionamiento, los tipos de datos pasados por parámetros y un ejemplo de su uso.

2.4 Tecnologías web del lado del cliente

Cada vez van surgiendo nuevas tecnologías que se relacionan con el desarrollo web lo que hacen a éste más interactivo y apasionante. Entre las tecnologías utilizadas, están las que funcionan del lado del cliente y del lado del servidor. Las del lado del cliente están insertadas en la página HTML del cliente y son interpretadas y ejecutadas por el navegador. Es decir, que su correcta funcionalidad depende del soporte de la versión del navegador a ser utilizado por el usuario visitante.

2.4.1 ExtJs

Es un framework basado en JavaScript Orientado a Objeto (JOO), uno de los más usados en el desarrollo de aplicaciones web en la parte del cliente, debido a que cuenta con una gran cantidad de componentes, clases y validaciones que pueden ser configuradas según las necesidades de cada sistema o del cliente.

Cuenta con una ayuda en inglés, la que tiene diversos ejemplos, de los cuales se pueden auxiliar los desarrolladores para realizar las aplicaciones. Tiene un diseñador de interfaces en línea, lo que facilita la creación de ventanas y componentes, garantizando un rápido desarrollo de sistemas web. (ExtJS development Group, 2010).

ExtJS no es solamente otro framework JavaScript, debido a que se puede integrar con otras librerías como la YUI de Yahoo, ampliando las funcionalidades. Con este framework se pueden desarrollar aplicaciones web de forma fácil y rápida debido a:

- Provee una forma fácil de uso de Document Object Model (DOM¹³), compatible con las ventanas, formularios y otros componentes.
- Maneja de forma transparente los distintos navegadores web, facilitando y agilizando el trabajo al programador, al no tener que definir varias Hojas de Estilo (CSS) en dependencia del navegador del cliente.
- Las peticiones y la comunicación con el servidor es usando AJAX¹⁴, por lo que no es necesario recargar la página, garantizando mayor velocidad en las peticiones y respuestas.
- El uso del DOM se realiza de forma transparente para el programador, debido a que ExtJS cuenta con una capa de abstracción que soporta a los navegadores más usados: Internet Explorer 6 o superior, Mozilla Firefox 2 o superior, Safari 2 o superior, Opera 9.0 o superior.

¹³ **DOM:** Interfaz de programación de aplicaciones.

¹⁴ **AJAX:** Técnica de desarrollo web para crear aplicaciones interactivas. Es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

La biblioteca ExtJS usa la notación JSON (del inglés JavaScript Object Notation) para el manejo de los datos y la configuración de los componentes, usando esta ventaja se le agregó la funcionalidad de carga en tiempo de ejecución de código JavaScript, donde se encuentran definidos los componentes a usar de ExtJS; logrando que el sistema sea más rápido en el proceso de carga. (Frederick, y otros, 2008).

2.4.2 OpenLayer

Es una biblioteca JavaScript desarrollada inicialmente por MetaCarta¹⁵ en Junio de 2006, la cual se mantiene como desarrolladora principal y ofreciendo soporte, desde noviembre del 2007 este proyecto forma parte de la Open Source Geospatial Foundation (OSGeo¹⁶). Este software libre permite mostrar un mapa dinámico y marcadores cargados desde cualquier fuente de datos, utilizando una variante de la licencia BSD¹⁷, destinada a mostrar mapas interactivos en los navegadores web. OpenLayers ofrece una API¹⁸ para acceder a múltiples tipos de fuentes de información cartográfica como Servicios de Mapas Web, Mapas comerciales, Servicios de Características Web, distintos formatos vectoriales, mapas de Open Street Map y otros.

Es un API para construir Mapas en aplicaciones web's, una biblioteca JavaScript que nos permite elaborar nuestros mapas haciendo uso de su propia base de información cartográfica (algo más pobre que la de Google Maps) o usar su estructura para hacer uso de otros servicios. (Barreda, 2007). OpenLayers permite la facilidad de colocar un mapa dinámico en cualquier página web. Se puede mostrar bloques de mapa y

¹⁵**MetaCarta:** Proveedor líder de soluciones geográficas inteligentes. Es una tecnología única que combina la búsqueda geográfica y las capacidades de etiquetado geográfico para que los usuarios puedan buscar contenido acerca de un lugar en un mapa.

¹⁶**OSGeo:** Organización no gubernamental cuya misión es dar soporte y promover el desarrollo colaborativo de tecnologías geoespaciales y datos abiertos. Persigue objetivos más allá del desarrollo de software, tales como promover el acceso libre a los datos geoespaciales gubernamentales y a los datos completamente libres.

¹⁷**Licencia BSD:** Licencia de software libre, tiene menos restricciones, permite el uso del código fuente en software no libre.

¹⁸**API:** Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

marcadores cargados de cualquier fuente. Se ha desarrollado para promover el uso de la información geográfica de todo tipo. Es totalmente gratuito, de open source JavaScript, publicada bajo una licencia tipo BSD¹⁹. Entre las ventajas que posee que fundamentan su uso se encuentran:

- No requiere instalación.
- Menor procesamiento en el servidor.
- Puede ampliar fácilmente el código para su aplicación en particular.
- Puede utilizar múltiples servidores de datos.

Un requisito que puede ser de desventaja en otro ambiente es que el desarrollador necesita de conocimientos de JavaScript, CSS y HTML. Para contrarrestar esto el uso de GWT²⁰ abstrae al desarrollador de dichos lenguajes encapsulándolos mediante el plugin GWT-OpenLayers en el lenguaje de desarrollo Java creando así una homogeneidad en el proceso de desarrollo. (Barreda, 2007).

Es fácil de instalar / usar. Es libre de utilizar y modificar el código fuente.

2.4.2.1 Herramientas disponibles

Cambiador de Capas: El control para la gestión de la visibilidad de capas.

Barra de zoom y navegación: Crea una barra de zoom y un panel de navegación que contienen los botones de Acercar y Alejar a los extremos. Hay tres maneras para utilizar esta barra:

- Clic sobre los botones de + o - puestos en los extremos de la barra. El resultado será el aumento o disminución de un nivel de zoom.
- Clic en cualquier punto de la barra, el desplazado (slider) se mueve al nivel de zoom más cercano. El mapa se actualiza al nivel de zoom correspondiente.

¹⁹**BSD:** La licencia de software libre permisiva, tiene menos restricciones en comparación con otras, permite el uso del código fuente en software no libre.

²⁰**GWT:** Framework creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX. Es compatible con varios navegadores.

- Desplazamiento del cursor a un nivel de zoom concreto cuando se deja el botón del ratón, el desplazado se posiciona en el nivel de zoom más cercano y el mapa se actualiza al nivel de zoom correspondiente.

Mapa de navegación: Crea un pequeño mapa de navegación. Este mapa enseña su posición principal y ofrece una herramienta más de navegación. Normalmente se posiciona en el ángulo inferior derecho y puede ser reducido con un clic en el botón que tiene en su borde. El rectángulo dibujado en este mapa puede ser movido para cambiar la posición del mapa principal.

Enlace: Ofrece un enlace al mapa. Esto permite poder guardar en un simple enlace el estado de la navegación del mapa. Así, se puede guardar como favorito en el navegador o enviarlo por correo.

Barra de herramienta del ratón: Es una simple barra de herramientas que permite escoger cómo utilizar el ratón entre navegación y zoom con ventana.

Línea a escala: Enseña una escala gráfica sobre el mapa.

Posición del ratón: Enseña las coordenadas actuales del cursor sobre el mapa. (Barreda, 2007).

2.4.3 GeoExt

La biblioteca GeoExt es una herramienta utilizada para construcción de aplicaciones web de mapas, basada en JavaScript. Reúne a los usuarios geoespaciales y los ubica en el cómo saber de OpenLayers con la interfaz de usuario de ExtJS para ayudarle a crear potentes aplicaciones SIG en la web con JavaScript. (Enmanuel, 2011). Es la biblioteca que integra el uso de Open Layer con Ext, utilizada por la facilidad en el desarrollo de la creación de mapas.

2.4.4 Estándar de codificación para Ext

Los ficheros JavaScript no cuentan con un estándar de codificación definido, debido a que al publicar una aplicación en modo de producción estos ficheros son comprimidos usando un compresor de código lo que convierte al código JavaScript ilegible, imposibilitando su modificación, pero acelerando la carga y ejecución de estos. Teniendo en cuenta lo anteriormente planteado se propone el siguiente estándar de

CAPÍTULO 2. HERRAMIENTAS Y TECNOLOGÍAS

codificación a usar en los ficheros con JavaScript vinculados a la biblioteca ExtJS, basado en JOO:

- Se debe usar 4 espacios para separar el código.
- Se deben definir las variables a usar dentro de la configuración de la propia variable.
- El comentario de línea se debe definir en una sola línea comenzando por el signo de comentario seguido de un espacio y el comentario en letra minúscula.
- Las variables string se deben definir entre comilla simple.
- Se debe usar la estructura de JOO para definir los objetos, entre llaves.
- Se debe usar la estructura de JOO para definir arreglos de objetos, entre corchetes.
- Las variables reservadas del lenguaje se deben escribir en minúscula, tales como: true, false, null, new, etc.
- En caso de que se tenga que escribir un comentario de varias líneas, se debe utilizar varios comentarios de línea.

2.5 Servidores

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). (ASF, 2010). A continuación se describen los utilizados.

2.5.1 Apache

Es un servidor web que se encarga de atender las solicitudes realizadas por los usuarios mediante la Internet, brinda la funcionalidad de publicar sitios en línea. El proyecto de Apache es creado y actualizado por la Apache Software Foundation (ASF). Es una organización no lucrativa, ya que distribuyen el servidor apache de forma libre para todo aquel que desee usarlo. ASF es una comunidad descentralizada de desarrolladores que trabajan cada uno en sus propios proyectos de código abierto.

Se fundó en 1999 a partir de la creación del Grupo Apache, el cual tenía como objetivo la creación de un servidor Http de código abierto. Entre los objetivos principales de la ASF está la de dar protección legal a todos los voluntarios que trabajan en “Proyectos

Apaches”. El Proyecto Apache es el origen de un grupo de licencias de código abierto, las cuales se denominan “Estilo Apache”. (ASF, 2010).

2.5.2 Postgres

Es un potente sistema de base de datos objeto – relacional de código abierto. Cuenta con un historial de 15 años de activo desarrollo, en los cuales se ha ganado una gran reputación por ser confiable, correcta integración de datos y alto rendimiento. Se puede ejecutar sobre la mayoría de los Sistemas Operativos (SO) que existen en la actualidad, como Linux, Microsoft Windows, Mac OS X, Solaris y Fedora. Cuenta con un completo soporte para llaves, llaves foráneas, uniones, vistas, disparadores y procedimientos almacenados. Tiene todos los tipos de datos que son usados por SQL Server 2008 y cuenta con interfaces para su comunicación en la mayoría de los lenguajes existentes como C, C++, C#, Perl, PHP, Java, Python, Ruby, ODBC, etc. (PostgreSQL Foundation, 2009).

Mediante el uso de las clases de PostgreSQL se cuenta con un avanzado compendio de opciones, tales como, Control Concurrente de Multi – Versión (MVCC), puntos de recuperación por tiempo, replicación asincrónica, puntos de salvamento, permite realizar respaldos de forma local o en línea, según la configuración del servidor. Cuando se activa el servidor postgres en ambiente de desarrollo avanzado se cuenta con una capacidad de almacenaje de 4 Tera Byte (TB), pero en modo de producción los datos de rendimiento y almacenaje son los siguientes:

- Capacidad máxima por tabla 32 TB.
- Capacidad máxima por fila 1.6 TB.
- Capacidad máxima por campo 1 TB.
- Ilimitada cantidad de filas por tabla.
- Cantidad de columnas por tabla entre 250 – 1600, en dependencia de los tipos de datos de cada columnas.
- Ilimitado indexado por tabla.

2.5.3 Postgis

Postgis ha sido desarrollado por la empresa canadiense RefractionResearch, especializada en productos open source entre los que habría que citar a Udig. Es hoy

día un producto veterano que ha demostrado versión a versión su eficiencia. En relación con otros productos, Postgis 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. (Martínez, 2010).

Un aspecto que tenemos que tener en cuenta es que Postgis ha sido certificado en 2006 por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas también interoperables. Almacena la información geográfica en una columna del tipo GEOMETRY, que es diferente del homónimo "GEOMETRY" utilizado por PostgreSQL, donde se pueden almacenar la geometría en formato WKB (Well-Known Binary), aunque hasta la versión 1.0 se utilizaba la forma WKT (Well-Known Text). (Martínez, 2010).

Se utiliza en el desarrollo de Atlas porque es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en SIG. Se publica bajo la Licencia pública general de GNU. Se explota porque es la base de dato más utilizada en la creación de SIG.

2.6 Herramientas

Es el instrumento de trabajo del desarrollador, en el presente epígrafe se describen las herramientas usadas en el desarrollo de la aplicación web.

2.6.1 Visual Paradigm

Es una poderosa herramienta CASE que al igual que el Rational Rose utiliza UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Genera código para varios lenguajes. Tiene integrado el MS Visio y es compatible con otras ediciones. Posibilita un entorno de creación de diagramas para UML 2.0. (Visual Paradigm Design Group, 2009).

A partir de un modelo relacional en SQL Server, MySql, entre otros, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Para gestionar la persistencia y el mapeo de estas clases con la

base de datos utiliza Hibernate para Java y NHibernate en el caso de un proyecto .Net. Es colaborativa, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como web o PDF, y permite control de versiones.

Visual Paradigm es una herramienta profesional. Permite crear todos los tipos de diagramas necesarios para demostrar los procesos de desarrollo. Brinda una serie de facilidades por lo que ha sido seleccionado de antemano por el arquitecto principal de la Plataforma de Gestión de Servicios para realizar el modelado de los diferentes diagramas.

Se utiliza por ser una herramienta UML profesional muy potente que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Diseñado para varios tipos de usuarios, incluyendo Ingenieros de Software, Analistas de Sistemas, Analistas de Negocio, Arquitectos de Sistema y Desarrolladores.

2.6.2 Zend Studio

Zend Studio es un IDE usado para el desarrollo de aplicaciones web, tanto para lenguaje PHP como para Java. Al ser creado y mantenido por el mismo equipo que desarrolla PHP, está optimizado para la integración con este lenguaje de programación. Está diseñado para aumentar la productividad, velocidad y rendimiento en el trabajo colectivo e individual. (Zend GDE, 2010).

Entre las funcionalidades que brinda está la refactorización de métodos y variables, generación dinámica de código lo que aumenta el desarrollo de las aplicaciones, cuenta con un analizador semántico, entre otros elementos. En conjunto todas estas funcionalidades permiten y garantizan el trabajo eficiente, rápido y seguro de los sistemas web.

Cuenta con la posibilidad de añadir más funcionalidades, mediante la inclusión de plugins, uno de los más usados en el desarrollo en equipo es el subclipse, el cual permite la conexión con un repositorio o control de versiones para asegurar un correcto trabajo en equipo. Cuenta con soporte y reconocimiento de la última versión de PHP y Eclipse Galileo usado para la programación en ficheros JavaScript, hojas de

estilo y páginas clientes. Tiene un amplio completamiento de código para la web y la posibilidad de integrarles las librerías que sean usadas en el desarrollo de proyectos.

2.7 Conclusiones parciales

En este capítulo se realizó un estudio de las tecnologías a utilizar en el desarrollo de la propuesta de solución así como algunos conceptos y tendencias que se deben tener en cuenta. Se fundamentó la elección del lenguaje de programación, el sistema gestor de bases de datos, y la metodología de desarrollo de software; así como el uso de otras técnicas y tecnologías.

Por todo lo expuesto en este capítulo, para la elaboración de la propuesta de solución que plantea esta investigación se desarrollará una aplicación web, lo que permite que el sistema pueda ser utilizado por todos los usuarios sin más requerimientos que una computadora con navegador web y conexión a la red.

Para el desarrollo de la aplicación se utilizará como metodología de desarrollo Agile UP (AUP), un enfoque simple y fácil de entender para el desarrollo de software usando técnicas y conceptos que aún se mantienen vigentes en RUP, se manipulará el IDE Zend Studio para la implementación del sistema ya que es una gran ventaja, debido a que cuenta con varias comodidades que garantizan que los desarrolladores realicen sus programas de forma rápida y óptima.

Capítulo 3. Análisis y Diseño.

3.1 Introducción.

En el presente capítulo se describen los requisitos funcionales y no funcionales que debe tener el sistema que se propone, lo que nos permite comprender el sistema de forma general, e identificar mediante un Diagrama de Caso de Uso las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan. Se mostrarán algunos diagramas de secuencia que servirán de apoyo para la comprensión del funcionamiento del sistema.

3.2 Descripción del Modelo de Dominio

Se muestra en la figura 3 que los usuarios que interactúan son el administrador del sistema y un usuario estándar, estos acceden según su rol en el sistema a los diferentes mapas que estén disponibles en el mismo. Un mapa está compuesto por varias capas y cada una de estas por un conjunto de elementos de una misma geometría, es decir, un mapa lo componen varias figuras diferentes como puntos, líneas y polígonos, agrupadas por capas independientes que están compuestas por elementos de un mismo tipo.

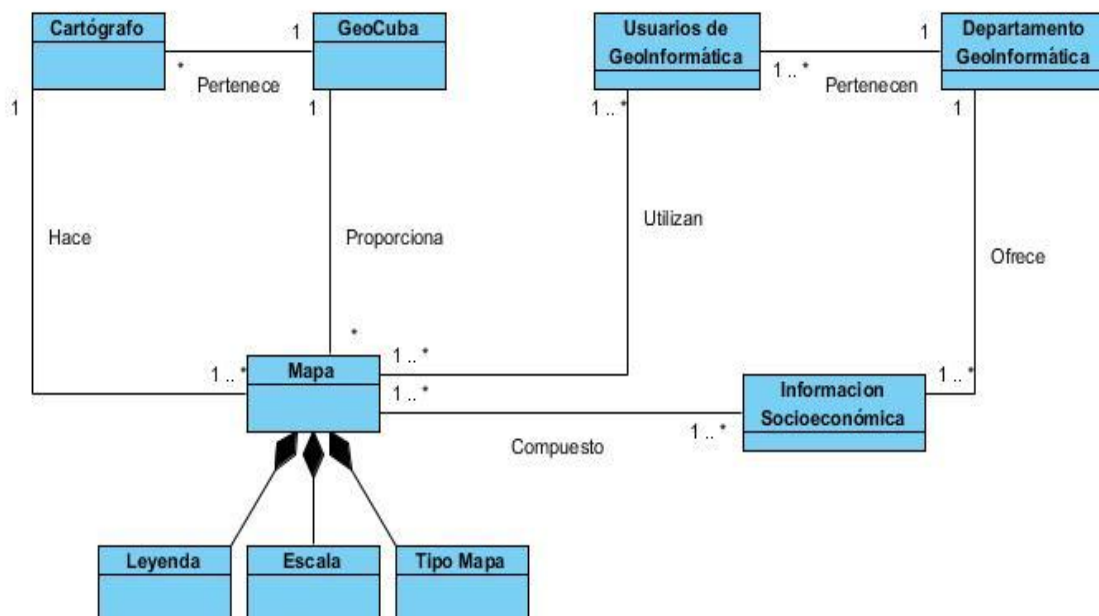


Figura 3 . Modelo de dominio.

3.3 Especificación de los requisitos de la aplicación

3.3.1 Requisitos funcionales

RF 1. Realizar Acercamiento: Con este requisito se permite aumentar el tamaño del mapa y disminuye la escala, ubicando en el centro del mapa el punto en el que el usuario realizó la operación de Acercar.

RF 2. Realizar Alejamiento: Con este requisito se disminuye el tamaño del mapa y aumenta la escala.

RF 3. Recentrar mapa: Con este requisito se permite que el usuario pueda recentrar el mapa al dar clic, logrando obtener la imagen a su máxima escala.

RF 4. Mover el mapa: Con este requisito se permite que el usuario pueda mover el mapa variando con el puntero del ratón la posición de la vista que se presenta.

RF 5. Localizar por Coordenadas: Con este requisito se activa una ventana donde el usuario puede introducir la longitud y latitud, y se ubica en pantalla el lugar de interés.

RF 6. Medir distancia: Con este requisito se quiere que el usuario pueda calcular la distancia entre dos o más puntos a partir del trazado de los mismos con el ratón y visualizar el valor de la distancia entre los dos últimos vértices dibujados y la distancia en un panel lateral.

RF 7. Calcular área: Con este requisito se quiere que el usuario pueda calcular el área de una región a partir del trazado de la misma con el ratón sobre el mapa, se visualizan los valores del área en un panel lateral.

3.3.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener y que harán del producto un sistema confiable y seguro.

➤ Apariencia o interfaz externa

El acceso a la aplicación y su presentación será sencillo, fácil de manejar por el usuario. Tendrá una interfaz amigable y similar al ambiente de desarrollo del sistema operativo Windows. Poseerá el color azul fundamentalmente.

Interfaces de usuario:

El sistema debe:

- Tener una apariencia profesional y un diseño gráfico sencillo.
- Posibilitarle al usuario la configuración del entorno de trabajo.
- Ser intuitivo.

Interfaces de hardware.

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 1 GB de memoria RAM.
- Procesador 512 MHz como mínimo.

Para los servidores:

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

➤ **Usabilidad**

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Se emplearán componentes que indiquen al usuario el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable.

El sistema tendrá una correcta arquitectura de la Información, a partir de un estudio de usuarios para su etiquetado e indexado.

Las funcionalidades principales del sistema estarán orientadas a iconos para un mayor reconocimiento por parte del usuario.

➤ **Fiabilidad**

La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores. La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.

Debido a la arquitectura que presenta el sistema, siendo más robusto al no tratarse de un sistema de gestión que requiera mantenimiento y optimización en el almacenamiento, se estima un tiempo promedio de 6 meses entre posibles fallas.

El tiempo medio de reparación, en caso de un fallo es de 7 días.

3.4 Definición de los actores y casos de usos del sistema

3.4.1 Actores del sistema

Un actor es una idealización de una persona externa, de un proceso, o de un objeto que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor. (Pressman, 2000).

En otras palabras se puede decir que un actor del sistema va a hacer todo aquel trabajador del negocio que posea actividades a automatizar como se muestra en la Tabla 2:

Actor	Descripción
Usuario	Cualquier usuario que utiliza las funcionalidades del sistema.
Sistema	El propio sistema.

Tabla 2 Descripción total del actor del sistema.

3.4.2 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. (Jacobson, 2000).

En la Figura 3, se muestra el diagrama de casos de uso del sistema para la plataforma de desarrollo de sistemas de información geográfica:

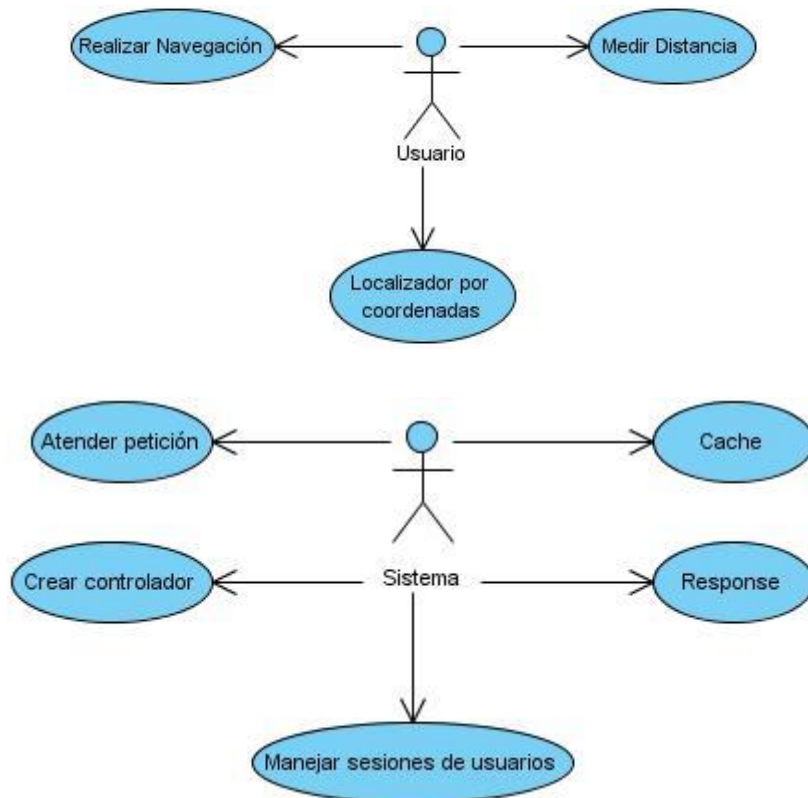


Figura 4 Diagrama de Caso de Uso del Sistema.

3.4.3 Especificación de los Casos de Uso del Sistema

Un caso de uso es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores. El propósito de un caso de uso es definir una pieza de comportamiento coherente, sin revelar la estructura interna del sistema.

La definición de un caso de uso incluye todo el comportamiento que implica: las líneas principales, las diferentes variaciones sobre el comportamiento normal, y todas las condiciones excepcionales, que pueden ocurrir con tal comportamiento, junto con la respuesta deseada. Desde el punto de vista de los usuarios, éstas pueden ser situaciones anormales. Desde el punto de vista de los sistemas, son las variaciones adicionales que deben ser descritas y manejadas. (Rumbaugh, 2000).

3.4.3.1 Realizar Navegación

Caso de Uso:	Realizar Navegación.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de modificar la visualización inicial del mapa en la pantalla.
Resumen:	El caso de uso se inicia cuando el usuario desea mover, ampliar, disminuir o recentrar el mapa y termina cuando el sistema muestra el mapa resultante en la pantalla.
Referencias	RF 1, RF 2, RF 3, RF 4
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicia cuando el usuario selecciona una de las opciones de navegación :</p> <ul style="list-style-type: none"> - Acercar - Alejar. - Recentrar Mapa. 	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> -Se seleccionó “Acercar”, ver sección “Acercar”. -Se seleccionó “Alejar”, ver sección “Alejar”. -Se seleccionó “Recentrar mapa”, ver sección “Recentrar mapa”.
	<p>3. El sistema procesa la información según la acción realizada por el usuario y actualiza la visualización del mapa.</p>

Prototipo de Interfaz



Sección “Acercar”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Acercar y dibuja un punto con un clic en la región específica del mapa que desea visualizar.	2. El sistema realiza la operación de Acercar seleccionada por el usuario aumentando el zoom y disminuyendo la escala puntualmente, es decir, admite las coordenadas (x, y) y mueve al centro del mapa el punto donde dio clic disminuyendo la escala.
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.
<p style="text-align: center;">Prototipo de Interfaz</p> 	
Sección “Alejar”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción del Alejar y marca un punto en el mapa.	2. El sistema realiza la operación de Alejar seleccionada por el usuario disminuyendo el zoom y aumentando la escala puntualmente, es decir, admite las coordenadas (x, y) y mueve al centro del mapa el punto donde dio clic aumentando la escala.

3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.

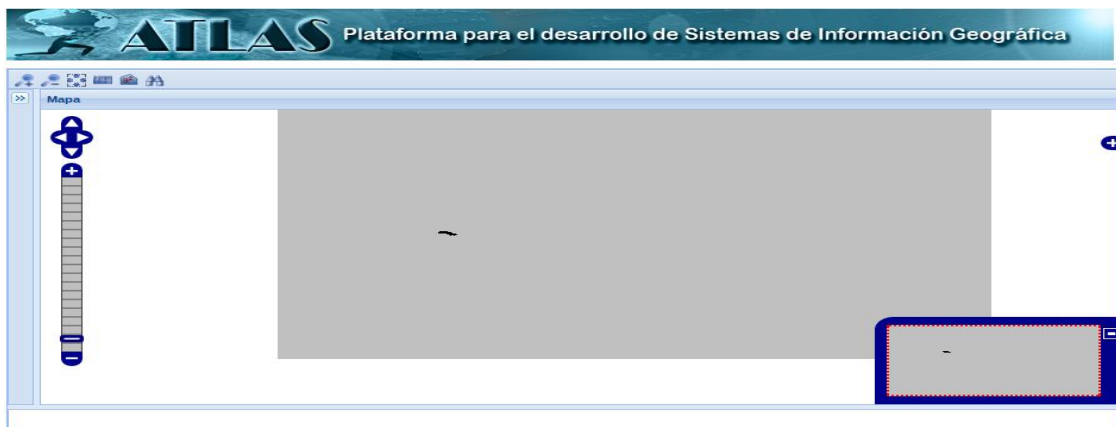
Prototipo de Interfaz



Sección “Recentar Mapa”

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Recentar el mapa y marca un punto en el mapa de la región que desea recentar.	2. El sistema obtiene las coordenadas (x, y) del punto donde el usuario hizo clic y lo traslada al centro de la pantalla, obteniendo al mapa en su máxima escala.
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.

Prototipo de Interfaz



Sección “Mover Mapa”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona da clic en un punto determinado del mapa y arrastra el mouse.	2. El sistema calcula un ΔX y un ΔY a partir del centro de pantalla y se obtiene un nuevo centro de pantalla a donde se movería el mapa. No varía la escala de representación del mapa.
	3. El sistema procesa la información a partir de la acción realizada por el usuario y actualiza la visualización del mapa.

Tabla 3 CUS Realizar Navegación.

3.4.3.2 Localizador por Coordenadas.

Caso de Uso:	Localizador por Coordenadas.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de localizar cualquier objeto del mapa de acuerdo a las coordenadas (x, y) proporcionadas por el usuario.
Resumen:	El caso de uso se inicia cuando el usuario desea localizar un objeto en el mapa por sus coordenadas, introducidas por él y termina cuando el sistema muestra el objeto redimensionando el mapa a las coordenadas introducidas.
Referencias	RF 5.
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 3. ANÁLISIS Y DISEÑO

1. El caso de uso se inicia cuando el usuario selecciona la opción “Localizar por Coordenadas”.	2. El sistema muestra la ventana Localización por Coordenadas. Para los valores de las “Coordenadas se tienen dos Cajas de texto, uno para la “Latitud” y otro para la “Longitud”.
3. El usuario introduce los datos asociados a los valores de coordenadas. Al finalizar da clic en el botón “Aceptar”.	4. El sistema realiza las conversiones necesarias para el zoom y redimensiona el mapa moviendo al centro de la pantalla las coordenadas introducidas por el usuario y marcando el objetivo con una tachuela.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
3.1 El usuario introduce los datos asociados a los valores de coordenadas y da clic en el botón “Aceptar”.	4.1 El sistema redimensiona el mapa moviendo al centro de la pantalla las coordenadas introducidas por el usuario.
Poscondiciones	El sistema visualiza en el centro del mapa las coordenadas introducidas por el usuario.

Tabla 4 CUS Localizador por Coordenadas.

3.4.3.3 Medir Distancia.

Caso de Uso:	Medir Distancia.
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de medir las distancias entre puntos en el mapa o un área determinada.
Resumen:	El caso de uso se inicia cuando el usuario desea medir la distancia existente entre 2 o más puntos dibujados en el mapa, y termina cuando el sistema muestra los valores correspondientes.
Referencias	RF 7, 8
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el usuario selecciona la opción <ul style="list-style-type: none"> ➤ “Calcular Distancia”. ➤ “Calcular área”. 	2. El sistema realiza la operación según la opción seleccionada por el usuario. <ul style="list-style-type: none"> -Se seleccionó “Calcular Distancia”, ver sección “Calcular Distancia”. -Se seleccionó “Calcular área”, ver sección “Calcular área”.
3. El usuario realiza el trazado de una poli línea dando clic izquierdo y arrastrando el cursor y termina con doble clic en el mapa cuando haya realizar el trazo deseado.	4. El sistema muestra en un costado el resultado, las distancias se expresan en la unidad de medida que muestra la aplicación por defecto.
	5. El sistema muestra en la barra de izquierda la distancia del último vértice al actual y la distancia total del primer vértice al actual.

Prototipo de Interfaz



Sección “Calcular Distancia”

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Calcular Distancia y dibuja un punto con un clic en la región específica del mapa que desea calcular, arrastrándolo hasta otro punto.	2. El sistema realiza la operación de Calcular Distancia seleccionada por el usuario.
	3. El sistema muestra en un costado el resultado, las distancias se expresan en la unidad de medida que muestra la aplicación por defecto.

Prototipo de Interfaz



Sección “Calcular Área”

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Calcular Área y dibuja un punto con un clic en la región específica del mapa que desea calcular, arrastrándolo hasta otro punto.	2. El sistema realiza la operación de Calcular Área seleccionada por el usuario.

3. El sistema muestra en un costado el resultado, las distancias se expresan en la unidad de medida que muestra la aplicación por defecto.

Prototipo de Interfaz

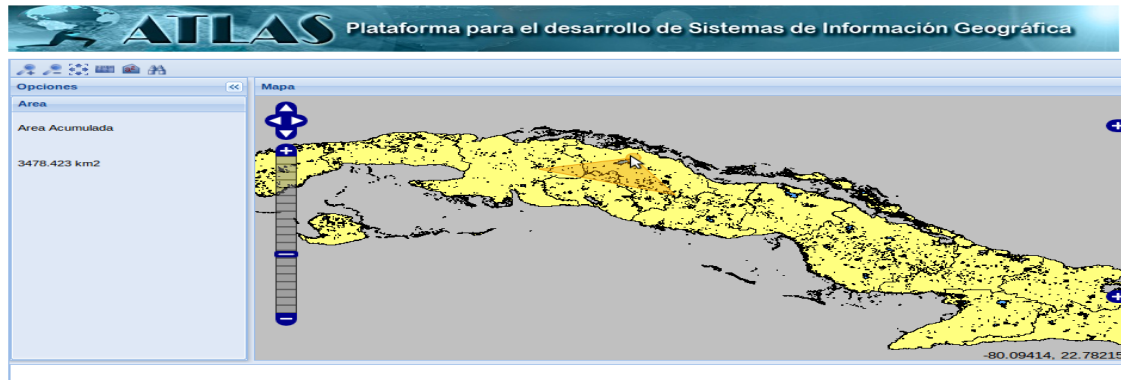


Tabla 5 CUS Medir Distancia.

3.5 Estilo arquitectónico utilizado

Teniendo en cuenta lo expuesto en el Capítulo 1 y 2 se utilizó para desarrollar la aplicación el patrón arquitectónico Modelo Vista Controlador. Este patrón de diseño se basa en la separación en tres capas del diseño de las aplicaciones: el modelo de datos, la prestación de los mismos y las acciones de los usuarios. Una ventaja de este patrón es que permite que desde la vista se pueda tener acceso a las clases del modelo, este es una gran ventaja en aplicaciones como esta, en las que se hace un uso intensivo de componentes visuales y requieren de una actualización constante de las propiedades que se modifican.

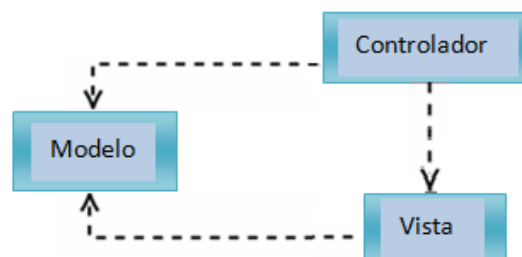


Figura 5 Diagrama Modelo Vista Controlador.

3.6 Patrones de diseños utilizados

Los patrones de diseño son soluciones estándares a problemas comunes en el desarrollo de software. En este epígrafe se describen los principales patrones de diseño utilizados en la solución de algunos problemas no triviales en la aplicación.

3.6.1 Patrón GRASP

Experto: Como la información está encapsulada en clases, cada una es responsable controlar el acceso a esa información.

Creador: Se crean objetos de una clase para acceder a la información de esta y poder manipularla.

Bajo acoplamiento: Hay poca dependencia entre las clases del sistema.

Alta cohesión: Para que ninguna clase maneje demasiada información, se ha distribuido la información.

3.6.2 Patrón SINGLETON

Se necesita mantener de manera única las instancias de los símbolos y de los adicionales en todo el tiempo de ejecución.

3.6.3 Patrón GOF

Recuerdo: Permite volver a estados anteriores del sistema.

3.7 Diagramas de clases del diseño

Un diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces de la aplicación. Sirve para visualizar las relaciones entre las clases que involucran el sistema. Ver anexo 1.

3.8 Diagramas de colaboración

Esencialmente es un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de comunicación como también puede llamarse, muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de comunicación no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.

- Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común.
- Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dicha implementación es llamada "enlace".

Un diagrama de comunicación es también un diagrama de clases que contiene roles de clasificador y roles de asociación en lugar de sólo clasificadores y asociaciones. Los roles de clasificador y los de asociación describen la configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la comunicación.

Cuando se instancia una comunicación, los objetos están ligados a los roles de clasificador y los enlaces a los roles de asociación. El rol de asociación puede ser desempeñado por varios tipos de enlaces temporales, tales como argumentos de procedimiento o variables locales del procedimiento. Los símbolos de enlace pueden llevar estereotipos para indicar enlaces temporales. Para cada uno de los casos de usos se realiza el diagrama de colaboración. Ver anexo 2.

3.9 Conclusiones parciales

Con la realización de este capítulo se plasmó el resultado obtenido durante el flujo de trabajo análisis y diseño. Para esto se tomó como base el modelo de dominio y modelo del sistema y se le realizó una especificación más detallada de los casos de usos. En todos los casos se hace una explicación de forma general del diseño, los requisitos funcionales y no funcionales, los cuales unidos a las descripciones van a servir de base a los desarrolladores que implementarán el sistema en un futuro, esto sentó las bases para desarrollar todo el trabajo de manera organizada y poder dar cumplimiento a las expectativas del cliente.

Capítulo 4. Desarrollo de la solución y validación.

4.1 Introducción

A través de la realización de este capítulo se describe cómo se realizó la implementación de la aplicación en términos de componentes. Se describen algunos fragmentos de códigos, se exponen los principales resultados obtenidos a lo largo del trabajo, a la vez que se discute sobre la importancia y relevancia de los mismos. También en este capítulo es la construcción de los casos de pruebas para las principales funcionalidades, cumpliendo así con una de las tareas que plantea AUP para el rol de implementador.

4.2 Modelo de implementación

El Modelo de Implementación describe cómo los elementos del Modelo de Diseño son implementados en términos de componentes. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponible en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y la dependencia entre los componentes. (Garrido, 2004).

El diagrama de componentes va a mostrar las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, archivos, paquetes, bibliotecas cargadas dinámicamente o ejecutables. En el desarrollo de Atlas se tuvo en cuenta cinco casos de usos fundamentales del núcleo de la aplicación.

A continuación se muestra el diagrama de componentes del sistema desarrollado según el caso de uso.

4.2.1 CU: Atender petición

En el caso de uso atender solicitud se muestra al controlador frontal que necesita para su ejecución del routing, es decir de la clase de ruteo, de request para el manejo de la petición del usuario, de controler que no son más que los controles específicos según la petición del usuario y a su vez controlador frontal, tratado como otro caso de uso la creación del mismo, también de user siendo el objeto de clase global, se utiliza para tener información sobre un usuario, y por último de response que va a manejar la respuesta que hay que darle a la petición realizada.

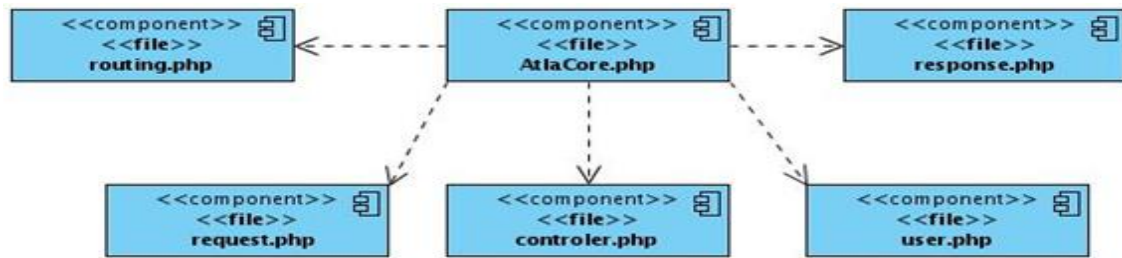


Figura 6 CU: Atender petición.

4.2.2 CU: Crear controlador

En el caso de uso crear controlador se debe tener en cuenta todo lo referente a las creaciones, el código js, la vista y las imágenes. Ver anexos 3.

4.2.3 CU: Manejar sesiones de usuarios

En el caso de uso de manejar sesiones de usuarios se necesita de la clase session, ya que es la que maneja todo lo relacionado con variables de sesión. Ver anexos 3.

4.2.4 CU: Cache

En el caso de uso cache se cuenta con la clase de lectura y escritura de ficheros de cache agilizando la carga de los mismos, el cual necesitará parsear en busca de ficheros según la extensión, a su vez también de config utilizándolo para la lectura, pero como los ficheros de configuración son .yaml se necesitará de yaml, ya que este lee y escribe dichos ficheros. Ver anexos 3.

4.2.5 CU: Response

En el caso de uso response se usa para manejar la respuesta que hay que darle a esa petición, haciendo uso del autoload, verificando que la clase esté incluida y de no estarlo la incluye, esto es cuando se crea un objeto de una clase con new. También se utiliza action para la llamada a un método de una clase. Ver anexos 3.

4.2.6 CU_Interfaz: Acercar

En el caso de uso acercar perteneciente a la interfaz, se verá reflejada su relación con las bibliotecas OpenLayer, ExtJs y a su vez con GeoExt que es la integración de ambas, además de su fichero js.

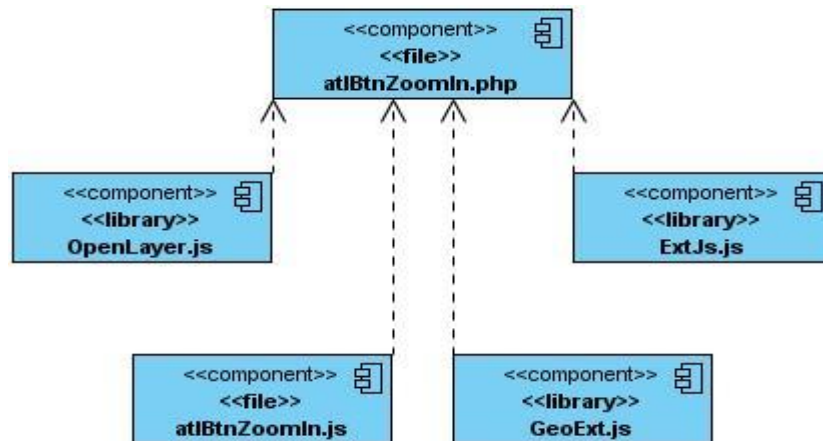


Figura 7 CU: Acercar.

Ver anexos 3 para algunas de las otras funcionalidades de la interfaz.

4.3 Modelo de despliegue

El modelo de despliegue se utiliza ya que es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El diagrama de despliegue para la aplicación que se desarrolla se muestra a continuación:

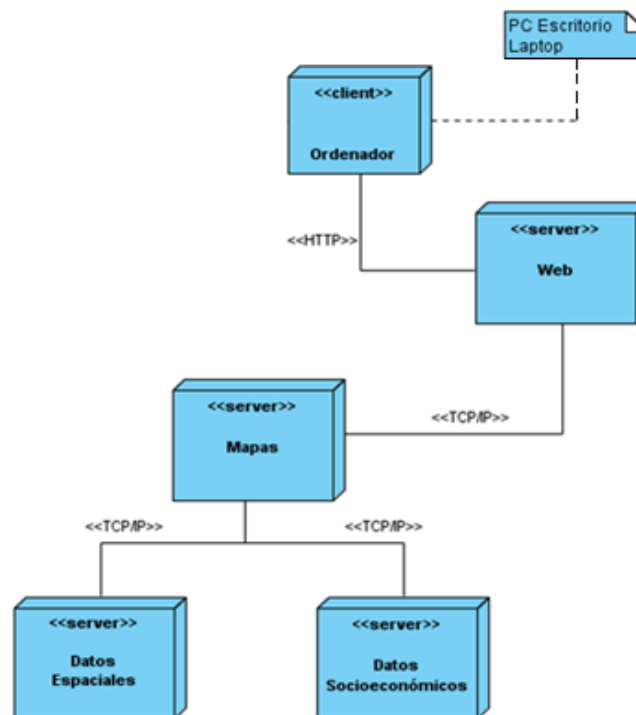


Figura 8 Diagrama de despliegue del sistema.

4.4 Pruebas unitarias

Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos. Las pruebas del software es un elemento crítico para la garantía de calidad del software.

Las pruebas unitarias verifican que cada método y función está trabajando correctamente, cada prueba deberá ser lo más independiente posible de las demás. Una de las buenas prácticas del desarrollo web que más cuesta a los programadores, consiste en realizar pruebas unitarias; debido a que no están acostumbrados a probar a profundidad los métodos y algoritmos diseñados. Las pruebas del software es un elemento crítico para la garantía de calidad del software.

Para las pruebas se escogió la técnica conocida como caja negra, que la misma se centrará en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas unitarias, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

4.4.1 Diseño de las pruebas que permitan validar la solución propuesta.

4.4.1.1 Realizar Navegación.

1. Descripción general

El caso de uso Realizar Navegación se inicia cuando el usuario desea mover, ampliar o recentar el mapa en dependencia de su selección, este es el CUS que trabaja directamente sobre el mapa, mostrando lo que el actor desea, y termina cuando el sistema muestra el mapa resultante en la pantalla.

2. Condición de ejecución

Ninguno.

CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN Y VALIDACIÓN

3. Descripción de variable

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Acercar	EC 1.1: Acercar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.
SC 2: Alejar	EC 2.1: Alejar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.
SC 3: Recentrar mapa	EC 3.1: Recentrar mapa cuando hace clic en un punto	El sistema actualiza la visualización del mapa.
SC 4: Mover Mapa.	EC 4.1: Mover Mapa cuando hace clic en un punto del mapa y arrastra.	El sistema actualiza la visualización del mapa.

Tabla 6 Secciones a probar en el caso de uso Realizar Navegación.

No existen variables para este Caso de Prueba.

4. Matriz de datos.

4.1 Acercar

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
C 1.1: Acercar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none">1. Seleccionar la opción Acercar.2. Dar clic en el mapa en la región que se desea visualizar.3. El sistema aumentará el zoom y disminuirá la escala puntualmente (coge las coordenadas (x, y) y se moverá al centro del mapa el punto donde se dio clic disminuyendo al doble la escala.

Tabla 7 Acercar.

CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN Y VALIDACIÓN

4.2 Alejar

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Alejar cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none">1. Seleccionar la opción Alejar.2. Dar clic en el mapa en la región que se desea visualizar.3. El sistema disminuirá el zoom y aumentará la escala puntualmente (coge las coordenadas (x, y) y se moverá al centro del mapa el punto donde se dio clic aumentando al doble la escala.

Tabla 8 Alejar.

4.3 Recentrar mapa

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Recentrar mapa cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none">1. Seleccionar la opción Recentrar Mapa.2. Dar clic en el mapa en la región que se desea recenter.3. El sistema obtiene las coordenadas del punto donde el usuario hizo clic y traslada al centro de la pantalla el punto seleccionado sin modificarse la escala del mapa.

Tabla 9 Recentrar mapa.

CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN Y VALIDACIÓN

4.4 Mover mapa.

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Mover mapa cuando hace clic en un punto del mapa.	El sistema actualiza la visualización del mapa.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none">1. Seleccionar la opción Mover Mapa.2. Dar clic y arrastrar hasta obtener la región del mapa que se desea visualizar.3. El sistema calcula un ΔX y un ΔY a partir del centro de pantalla y se obtiene un nuevo centro de pantalla a donde se movería el mapa.

Tabla 10 Mover mapa.

4.4.1.2 Localizar por Coordenadas.

1. Descripción general

El caso de uso se inicia cuando el usuario desea localizar un objeto del mapa por sus coordenadas, introducidas por él y termina cuando el sistema muestra el objeto redimensionando el mapa a las coordenadas introducidas.

2. Condición de ejecución

Ninguno.

3. Descripción de variable

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Localizar por Coordenadas	EC 1.1: Localizador por Coordenadas completo.	El sistema visualiza en el centro del mapa las coordenadas introducidas por el usuario.

Tabla 11 Secciones a probar en el caso de uso Localizar por Coordenadas.

CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN Y VALIDACIÓN

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1.	Longitud	Campo de texto	No	Se deben introducir datos y no dejar vacío el campo.
2.	Latitud	Campo de texto	No	Se deben introducir datos y no dejar vacío el campo.
3.	Vista actual	Campo de selección. (CheckBox)	Sí	Puede o no que este campo se seleccione.
4.	Zoom	Lista de Selección. (ComboBox)	Sí	Puede o no introducirse datos en este campo.

Tabla 12 Variables a probar.

4. Matriz de datos.

4.1 Localizador por Coordenadas completo.

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Localizador por Coordenadas completo.	El sistema visualiza en el centro del mapa las coordenadas introducidas por el usuario.	El sistema actualiza la visualización del mapa. Exitosa, cumplida.	<ol style="list-style-type: none"> 1. Seleccionar la opción Localizador por Coordenadas. 2. El sistema muestra una ventana para insertar los valores correspondientes. Para los valores de las "Coordenadas se tienen dos TextBox uno para la "Latitud" y otro para la "Longitud". Para "Modificar el nivel de zoom" se tiene un CheckBox que al marcarlo activa un ComboBox para seleccionar los valores del Zoom. 3. Llenar los valores de cada uno de los campos. 4. Dar clic en el botón "Aceptar". 5. El sistema visualiza en el centro del mapa las coordenadas introducidas.

Tabla 13 Localizar por Coordenadas completo.

CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN Y VALIDACIÓN

4.4.1.3 Medir distancia.

1. Descripción general

El caso de uso se inicia cuando el usuario desea medir la distancia existente entre 2 o más puntos marcados en un mapa, y termina cuando el sistema muestra los valores correspondientes.

2. Condición de ejecución

Ninguno.

3. Descripción de variable

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Medir distancia.	EC 1.1: Calcular distancia.	El sistema muestra el valor correspondiente a la distancia según la polilínea trazada por el usuario.
	EC 1.2: Calcular área.	El sistema muestra el valor correspondiente al área trazada por el usuario.

Tabla 14 Secciones a probar en el caso de uso Medir distancia.

No existen variables para este Caso de Prueba.

4. Matriz de datos.

4.1 Calcular distancia.

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.1: Calcular distancia.	El sistema muestra el valor correspondiente a la distancia según la polilínea trazada por el usuario.	El sistema actualiza la visualización del mapa mostrando el resultado. Exitosa, cumplida.	1. Seleccionar la opción Calcular Distancia. 2. Dibujar la polilínea. 3. Mostrar la distancia acumulada.

Tabla 15 Medir distancia.

CAPÍTULO 4. DESARROLLO DE LA SOLUCIÓN Y VALIDACIÓN

4.2 Calcular área.

Escenario	Respuesta del sistema	Resultado de la prueba	Flujo Central
EC 1.2: Calcular área.	El sistema muestra el valor correspondiente al área trazada por el usuario.	El sistema actualiza la visualización del mapa mostrando el resultado. Exitosa, cumplida.	<ol style="list-style-type: none">1. Seleccionar la opción Calcular área.2. Dibujar un área.3. Mostrar el área calculada.

Tabla 16 Calcular área.

4.5 Pruebas funcionales

Las pruebas funcionales se centrarán más en el código, debido a como su nombre lo dice, se enfocará en la funcionalidad como tal, se interesa porque en lo implementado obtenga el resultado esperado.

Verifican que la aplicación resultante se comporta correctamente en todo su conjunto. Las pruebas funcionales son la mejor opción para probar la aplicación de extremo a extremo, desde la petición realizada por un navegador hasta la respuesta enviada por el servidor. Son muy similares a lo que se hace manualmente: cada vez que añades o modificas una acción, se prueba en el navegador para comprobar que todo funciona bien al pulsar sobre los enlaces y botones y que todos los elementos se muestran correctamente en la página.

Debido al uso de ExtJS para el diseño de la interfaz de la aplicación, uso de carga dinámica en tiempo de ejecución de los ficheros con el código perteneciente a la interfaz y la inexistencia de una herramienta para probar (testear) peticiones Ajax, resulta muy difícil realizar las pruebas funcionales.

Teniendo en cuenta lo anteriormente planteado en caso de ser necesario validar alguna de las funcionalidades del sistema, ejemplo de esto se exponen dos códigos de dos de las funcionalidades del módulo de navegación, de acercar y alejar (ver anexo 4).

4.6 Conclusiones parciales

Como resultado de este capítulo se obtuvo el diagrama de componente, se explicaron algunas soluciones a nivel de código, se validan la mayoría de las funcionalidades, en caso de ser necesario se debe validar el sistema con una mayor profundidad. También se tuvo en cuenta los diagramas de implementación presentados, los cuales representan los casos de usos principales.

Conclusiones Generales

Una vez terminada la investigación, se evidencia que con la realización de la plataforma Atlas para el centro de desarrollo de software GEySED, específicamente para el polo de Geoinformática, se han cumplido los objetivos establecidos al inicio de la investigación, mejorando los procesos creación de los SIG.

El objetivo planteado en el diseño de la investigación fue cumplido, comprobándose la idea a defender como solución al problema científico que originó la investigación, tributando a los procesos.

Como resultado de la presente investigación, ya el centro cuenta con una aplicación web capaz de automatizar correctamente los procesos de creación de los SIG, la cual es capaz de manejar eficientemente los datos usados en los procesos mencionados.

Recomendaciones

Se recomienda realizar una segunda versión de la aplicación, teniendo en cuenta los aspectos señalados en esta, incorporar una ayuda al sistema, garantizando la usabilidad del mismo. Realizarle todas las pruebas necesarias. Seguir mejorando en la creación de aplicaciones o herramientas como estas para el desarrollo de SIG.

Bibliografía

Alfaro, F.M. 2005. *Tecnología Cliente - Servidor*. 2005.

Alfaro, Mg. Emigdio. 2007. *Desarrollo de Sistemas Web*. 2007.

Alvaro de J. Carmona, Jhon Jairo Monsalve R. 2009. *Sistemas de Información Geográfica*. 2009.

Angel Alvarez, Miguel. 2004. Que es la Programacion Orientada a Objeto. [En línea] 24 de Julio de 2004. [Citado el: 3 de Marzo de 2010.] <http://www.desarrolloweb.com/articulos/499.php>.

Angel Alvarez, Miguel y López, Daniel, Gutierrez,Manu. 2007. *PHP 5: Programacion Orientada a Objeto*. 2007.

ASF. 2010. Apache - Web Server. [En línea] 05 de 01 de 2010. [Citado el: 10 de 02 de 2010.] <http://www.apache.org/>.

Barreda, Jose. 2007. *OpenLayer*. Perú : s.n., 2007.

Ben Kiki, Oren, Evans, Clark y Ingerson, Brian. 2006. *YAML Ain't Markup Language*. 2006.

BOCHER, S. STEINIGER Y E. 2009. *Capaware: Plataforma de desarrollo de software, 3ra Jornadas de SIG Libre*. Girona : s.n., 2009.

CODIMA, Marc Rosés. 2004. *Plataformas para el desarrollo de aplicaciones SIG en internet por personal no especializado*. Brasil : s.n., 2004.

Enmanuel. 2011. *Sistemas de Información Geográfica*. 2011.

ExtJS development Group. 2010. ExtJS Javascript Framework and RIA Plataform. [En línea] 03 de Marzo de 2010. [Citado el: 08 de Marzo de 2010.] <http://www.extjs.com/>.

Frederick, Shea, Ramsay, Colin y Blades, Steve 'Cutter'. 2008. *Learning ExtJS*. s.l. : PACKT, 2008. ISBN 978-1-847195 .

- Garrido, J. S. C. 2004.** *Arquitectura y diseño de sistemas web modernos.* 2004.
- Gironés, Jordi. 2007.** *Puro Software.* Argentina : s.n., 2007.
- González, Pedro José. 2010.** *Implementación de los módulos de Facturación y Cobro del Sistema de Facturación y Cobro para la Empresa de Gas Manufacturado de Ciudad de la Habana.* Habana : s.n., 2010.
- Group, PHP. 2010.** *PHP Hipertext Preprocesor.* 2010.
- Guardado, Iván. 2010.** *Doctrine.* 2010.
- Guglielmetti, Marcos. 2005.** *Definición de Sistema web.* 2005.
- Jacobson, I., G. Booch, and J. Rumbaugh,. 2000.** *El Proceso Unificado de Desarrollo de Software.* 2000.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2004. ISBN 84-7829-036-2.
- Lobo, Armando Robert. 2006.** *Tutorial Básico de ExtJS.* 2006.
- Martínez, Jose Carlos. 2010.** *PostGIS. Bases de datos espaciales.* . España : s.n., 2010.
- Microsoft. 2009.** MSDN Home Page. [En línea] <http://msdn.microsoft.com/es-es/library/aa291591%28VS.71%29.aspx>. [En línea] 2009.
- Minas, Facultad de ingeniería de. 2009.** *Manuales ArcView.* Piura : s.n., 2009.
- Montalvo, Carlos. 2011.** *TCPDF.* Lima. Perú : s.n., 2011.
- Montesino, Noraisi Prat. 2009.** *Frameworks.* 2009.
- Oviedo, Diego Alpízar Naranjo e Iván Argüello. 2006.** *Introducción al Agile UP.* Costa Rica : s.n., 2006.
- Park, Amelia Industrial. 2008.** *GSInnova.* Puerto Rico : s.n., 2008.
- php.net. 2010.** *PHP Development.* s.l. : Philip Olson, 2010.
- Pacheco, Alberto. 2005.** *Plataforma Informática.* Habana.Cuba : s.n., 2005.

PostgreSQL Foundation. 2009. PostgreSQL - Database Server. [En línea] 15 de 12 de 2009. <http://www.postgresql.org/>.

Potencier, Fabien y Zaninotto, François. 2009. *Symfony, guía definitiva*. 2009.

Pressman, R.S. 2000. *Ingeniería del Software. Un enfoque práctico*. 2000.

Rivero Varona, Helimay. 2009. *Análisis y diseño del sistema de facturación y cobro*. Ciudad de la Habana : s.n., 2009.

Rumbaugh, J., I. Jacobson, and G. Booch. 2000. *El Lenguaje Unificado de Modelado. Manual de Referencia*. 2000.

Sánchez, Alain y Rodríguez, Darlenis. 2007. *Sistema de gestión para el Programa Nacional de Grupos Electrogenos DENYO*. Ciudad de la Habana : s.n., 2007.

Sendra, Bosque. 1992. *Sistemas de Información Geográfica*. Madrid : Rialp, 1992.

Visual Paradigm Design Group. 2009. Visual Paradigm. [En línea] 12 de Noviembre de 2009. [Citado el: 18 de 01 de 2010.] <http://www.visual-paradigm.com/>.

Zaldívar, Yoenis Pantoja. 2009. *Módulo de análisis de la Plataforma GENESIG*. Habana : s.n., 2009.

Zend GDE. 2010. Zend Studio Integration Development Environment. [En línea] 22 de Febrero de 2010. [Citado el: 25 de Febrero de 2010.] <http://www.zend.com/>.

Glosario de Términos

APLICACION WEB: En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador

CASE: Las herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

CU: Caso de uso.

CUS: Caso de uso del sistema.

GRASP: Es un acrónimo que significa General Responsibility Assignment Software Patterns (Patrones generales de software para asignar responsabilidades).

GOF: Patrón de diseño Gang of Four.

IDE: Entorno de desarrollo integrado.

SIG: Sistema de Información Geográfica.

SOFTWARE: Conjunto de instrucciones y datos codificados para ser leídas e interpretadas por una computadora. Estas instrucciones y datos fueron concebidos para el procesamiento electrónico de datos.

PATRÓN ARQUITECTÓNICO: Define la estructura de un sistema software, el cual a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema.

UCI: Universidad de las Ciencias Informáticas.

UML: Lenguaje Unificado de Modelado.