

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Infraestructura de Datos Espaciales para la Universidad de las Ciencias Informáticas”

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

Autores: Maidelyn Piñero González

Emilio Kubota López

Tutor: Ing. Yenier Jiménez Morales

Co-tutor: Ing. Pedro Enrique Palau Isaac

La Habana, junio de 2012
“Año 54 de la Revolución”



“Queremos que tengan el máximo de conciencia de su papel, de lo que pueden hacer por su país, de lo que pueden hacer por la Revolución, de lo que pueden hacer por su futuro”.

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Se declaran a Maidelyn Piñero González y Emilio Kubota López como únicos autores de este trabajo y se le concede al centro Geoinformática y Señales Digitales (GEySED) de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmamos la presente declaración de autoría a los 19 días del mes de junio del año 2012.

Maidelyn Piñero González

Emilio Kubota López

Firma del Autor

Firma del Autor

Ing. Yenier Jiménez Morales

Ing. Pedro Enrique Palau Isaac

Firma del Tutor

Firma del Co-Tutor

DATOS DE CONTACTO

Tutor: Ing. Yenier Jiménez Morales

Profesión: Ingeniero en Ciencias Informáticas.

Año de graduado: 2010

Correo electrónico: yimorales@uci.cu

Co-Tutor: Ing. Pedro Enrique Palau Isaac

Profesión: Ingeniero en Ciencias Informáticas.

Año de graduado: 2011

Correo electrónico: pepalau@uci.cu

AGRADECIMIENTOS

Primeramente quisiera agradecer a la Revolución por darnos la posibilidad de estudiar en esta universidad, a mi tutor Yenier y co-tutor Pedro por la gran ayuda y apoyo en la realización del trabajo de diploma, a aquellos profesores que posibilitaron que llegase hasta aquí y a los profesores del proyecto por tenerme paciencia, por guiarme y brindarme sus conocimientos, especialmente a Lilianne y Lisandra. Agradecerle a mi mami Migdalia por confiar en mí, ser mi amiga y darme su apoyo incondicional, a mi novio Adnier por levantarme cuando más lo necesitaba y creer en mí en todo momento, a mi familia, mis compañeros y amistades cercanos y lejanos que se preocuparon por mí y me apoyaron constantemente.

Maidelyn Piñero González

Agradezco especialmente a mi madre y a mi padre por ser mis guardianes en todo momento y darme las fuerzas y la confianza necesarias para construir mi futuro. A toda mi familia por el apoyo incondicional que siempre me han brindado, porque han sabido ser mi soporte cuando más lo he necesitado. A mis amigos, por ser las mejores personas que he conocido en mi vida y que han estado siempre a mi lado sin importar las circunstancias. Agradecimientos a mi tutor Ing. Yenier Jiménez Morales y a mi co-tutor Ing. Pedro Enrique Palau Isaac, además de todo el colectivo del proyecto Aplicativos_SIG por todos los momentos compartidos y el esfuerzo realizado para hacer posible este sueño. Agradezco a mis compañeros de casa por ser buenos colegas todo el tiempo.

Emilio Kubota López

DEDICATORIA

Quisiera dedicarle este título en especial a mi madre que es mi razón, mi guía y mi ejemplo en la vida. A mi padre Rolando y familiares que me apoyaron en todo momento. A mis hermanos Oscar, Damian y Marlon. A mi novio Adnier que llegó mi vida para llenarla de alegrías y para demostrarme que siempre que hay sacrificio, hay un resultado. A mi tutor y amigo Yenier. A mis compañeros de aula que vienen conmigo desde 1er año y los que fui conociendo en el transcurso de los años, con los que compartí grandes momentos y posibilitaron que llegara hasta la recta final tras 5 años de estudio, sacrificio, alegrías y tristezas. A amigos de por vida Daykeni, Lietter, Yaneslearser y Janeth.

Maidelyn Piñero González

A mi eterna novia, mi madre Rosa que ha sido mi luz en cada amanecer. A mi padre Emilio por ser mi ídolo. A mi abuelo Koichi Kubota Ogawa, por depositar toda su confianza en mí en todo momento cuando me acompañaba (EPD). A mi familia completa, por preocuparse siempre por mi bienestar. A mi hermanito pequeño que lo quiero mucho y a mi primo Daniel. A mis amigos, que son una parte muy importante en mi vida y que siempre me ayudaron, en especial a Viñolo, Juan Pablo, Milena, Liset, Jen, Ángel Antonio, Eli, Daniel Ibargüen, Daili, Paul, Ary Sánchez Medina, Javier Carrillo, los quiero mucho. Ya todos que de una forma u otra hicieron posible este anhelo.

Emilio Kubota López

RESUMEN

El desarrollo de las tecnologías en la actualidad ha potencializado con el uso de la información geográfica la automatización de actividades tales como la localización, medición de distancia, cálculo de áreas, implicando el uso de los mapas como factor importante en la toma de decisiones. Un Sistema de Información Geográfica, es una aplicación que hace uso de la información geográfica para brindar funcionalidades de análisis, manejo, consulta y visualización de los datos. La interoperabilidad entre varios sistemas, la centralización de la información geográfica, el uso de los metadatos y la realización de operaciones a través de servicios web eran problemas persistentes en Sistemas de Información Geográfica, dando lugar al surgimiento de un nuevo sistema que provee los elementos relacionados anteriormente, la Infraestructura de Datos Espaciales. El desarrollo de una Infraestructura de Datos Espaciales para la Universidad de las Ciencias Informáticas facilita el acceso, visualización, consulta y combinación de la información geográfica, metadatos y servicios web a través de su integración, provisión y centralización. La creación de un visor geográfico y un catálogo de servicios para la entidad, desarrollados como nuevos módulos en el CMS Drupal constituyen un aporte significativo que aseguran la interoperabilidad entre los sistemas a partir del establecimiento de normas establecidas. La Infraestructura de Datos Espaciales provee la disponibilidad de los datos geográficos, el uso de los metadatos como elementos descriptores de dichos datos, su publicación y consumo por medio de servicios creados.

Palabras claves: información geográfica (IG), Sistema de Información Geográfica (SIG), metadatos, servicios de mapas en la web, Infraestructura de Datos Espaciales (IDE), visor geográfico.

Índice de Contenido

DECLARACIÓN DE AUTORÍA	III
DATOS DE CONTACTO.....	IV
AGRADECIMIENTOS.....	V
DEDICATORIA	VI
RESUMEN.....	VII
CONTENIDO	VIII
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1. Introducción.....	6
2. Conceptos asociados al dominio del problema.....	6
2.1. Servidor de Mapa:.....	6
2.2. Información geográfica o dato geográfico:.....	6
2.3. Datos georreferenciados:	6
2.4. Datos temáticos:	6
2.5. Metadatos:	7
2.6. Interoperabilidad:	7
2.7. Servicios:	7
2.8. Sistemas de Información Geográfica (SIG):	7
3. Infraestructura de Datos Espaciales (IDE).....	7
3.1. Objetivos de la Infraestructura de Datos Espaciales.....	8
3.2. Servicios utilizados en la Infraestructura de Datos Espaciales	8
3.2. Análisis de las soluciones existentes.....	10
4. Propuesta de solución.....	13
5. Tecnologías utilizadas en el desarrollo del sistema	13

5.1.	Metodología de desarrollo de software: Proceso Unificado de Desarrollo	13
5.2.	Lenguaje de modelado: Lenguaje de Modelado Unificado	14
5.3.	Leguajes de Programación.....	15
5.4.	Arquitectura: Cliente-Servidor	16
6.	Herramientas a utilizar.....	17
6.1.	Librería de visualización de mapas: OpenLayers	17
6.2.	Librería para interfaz de usuario: jQuery	18
6.3.	Servidor de mapas: MapServer.....	19
6.4.	Entorno integrado de desarrollo: Netbeans 7.0	20
6.5.	Herramienta de modelado de software: Visual Paradigm	20
6.6.	Sistema de Gestor de Contenido: Drupal	21
7.	Conclusiones parciales del capítulo.....	22
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA		23
1.	Introducción.....	23
2.	Modelo de Dominio	23
2.1.	Descripción de las clases del dominio.....	24
3.	Requisitos Funcionales (RF)	25
4.	Requisitos No Funcionales (RNF)	26
4.1.	Usabilidad	27
4.2.	Fiabilidad y disponibilidad	27
4.3.	Eficiencia.....	27
4.4.	Soporte	27
4.5.	Apariencia o interfaz externa.....	27
4.6.	Hardware	27
4.7.	Software.....	28
4.8.	Seguridad.....	28

5.	Modelo de Casos de Uso del sistema	28
5.1.	Actores del sistema	29
5.2.	Descripción textual del Caso de Uso Gestionar Servicios de Mapas en la Web	30
6.	Conclusiones parciales del capítulo.....	37
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN		38
1.	Introducción.....	38
2.	Estilos arquitectónicos.....	38
2.1.	Estilo Llamada y Retorno.	38
3.	Arquitectura de Software	38
3.1.	Patrón Arquitectónico N – Capas.....	39
3.2.	Patrón Arquitectónico Modelo- Vista- Controlador (MVC)	40
4.	Patrones de Diseño.....	41
4.1.	Patrones GRASP	41
4.2.	Patrones GoF.....	42
5.	Vistas del sistema	43
5.1.	Vista Lógica.....	43
6.	Visión general de la arquitectura	44
7.	Modelo de diseño	45
7.1.	Diagrama de clases del diseño	45
8.	Modelo de datos.....	47
8.1.	Diseño de la base de datos.....	48
8.2.	Descripción de las tablas.....	48
9.	Vista de despliegue	48
9.1.	Diagrama de Despliegue.....	49
10.	Diagrama de componentes	49
11.	Estilos de Codificación	51

11.1.	Principios Generales	52
12.	Conclusiones parciales del capítulo.....	54
CAPÍTULO 4: VALIDACIÓN DEL SISTEMA		55
1.	Introducción.....	55
2.	Pruebas del sistema.....	55
3.	Diseño de caso de Prueba del CU Gestionar Servicio WMS	56
3.1.	Descripción General.....	56
3.2.	Condiciones de Ejecución.	56
3.3.	Secciones a probar en el Caso de Uso.	56
3.4.	Descripción de variable	58
3.5.	Matriz de Datos	59
3.6.	Resultados de las Pruebas.....	62
4.	Conclusiones Parciales	63
CONCLUSIONES		64
RECOMENDACIONES		65
BIBLIOGRAFÍA.....		66
GLOSARIO		68

Índice de Figuras

Figura 1: Modelo de Dominio	23
Figura 2: Diagrama de Casos de Uso del Sistema	29
Figura 3: Diagrama del Caso de Uso Gestionar Servicio WMS	30
Figura 4: Estructura de Drupal	39
Figura 5: Arquitectura de Drupal	39
Figura 6: Esquema del MVC	40
Figura 7: Diagrama Vista Lógica	44
Figura 8: Diagrama de Clase del Diseño Gestionar Servicio WMS	46
Figura 9: Diagrama de Entidad-Relación	48
Figura 10: Diagrama de despliegue	49
Figura 11: Diagrama de componentes del módulo ide	50
Figura 12: Diagrama de componente del módulo ide_ui	50
Figura 13: Diagrama de componente del módulo idevisor	51
Figura 14: Diagrama de componente del módulo ide_servie_catalog	51
Figura 15: Ejemplo de código	53
Figura 16: Resultados de las pruebas	63

Índice de Tabla

Tabla 1: Descripción del Caso de Uso Gestionar Servicio WMS	37
Tabla 2: Escenario del Caso de Uso Gestionar Servicio WMS	58
Tabla 3: Variables del Caso de Uso Gestionar Servicio WMS	58
Tabla 4: Matriz de datos de Adicionar Servicio WMS	60
Tabla 5: Matriz de datos de Modificar Servicio WMS	61
Tabla 6: Matriz de datos de Listar Servicio WMS	61
Tabla 7: Matriz de datos de Eliminar Servicio WMS	62

INTRODUCCIÓN

En la actualidad, con el constante desarrollo de Internet¹ comúnmente conocida como “Red² de redes”, se han realizado avances en la ciencia y la tecnología; propiciando que el hombre aumente su perspectiva y desee automatizar muchas de las actividades que diariamente efectúa para hacer su trabajo menos engorroso.

Localizar, visualizar y analizar la información atendiendo a variables geográficamente forma parte de la cotidianidad, de manera que representa una de las actividades más empleadas por el hombre. Se evidencia cuando necesita ubicarse en un lugar, encontrar algún país o una región, saber la diferencia de horarios, conocer la densidad de población, estudiar el relieve, el clima, los océanos, la vegetación de un terreno, medir distancias, calcular áreas, entre otras operaciones. Esto se facilita con el uso de mapas³ que ayuden a tomar las decisiones correctas. Es así que comienza la digitalización de la información geográfica dando paso a que en las redes de ordenadores se dispongan de un gran volumen de datos geoespaciales.

El procesamiento, la representación y búsqueda de la información geográfica se realizaba de forma manual, por lo que el trabajo se tornaba muy complejo, de ahí que surgen los Sistemas de Información Geográfica (SIG), como tecnología clave para su gestión. Un SIG es la integración de *hardware*⁴, *software*⁵, datos, herramientas y fuerzas de trabajo necesarias, que permiten el análisis, manejo, localización, visualización y consulta de los datos geográficos (1).

Debido al constante avance en la utilización de la información geográfica, surgieron nuevas ideas para brindar mejoras prácticas que los SIG no ofrecían. Entre ellas se encuentran la gestión de la información desde varias fuentes disponibles, la centralización de los datos y la interoperabilidad entre los sistemas para brindar datos y servicios a través de la red. Además la obtención de la información geográfica se dificulta pues los proveedores no poseen la correcta documentación o está incompleta. Esto provoca que se encuentre dispersa, no exista o es imposible consultarla.

Para ofrecer una solución a las deficiencias presentadas se crean las Infraestructuras de Datos Espaciales (IDE). Constituyen la evolución lógica de los SIG y pueden describirse como SIG implementados sobre la red. Se trata de una estrategia organizativa, asumida por normas que van a

¹**Internet o Ciberespacio:** Red que permite la interconexión descentralizada de computadoras.

²**Red:** Conjunto de equipos informáticos y software conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios.

³**Mapas:** Una representación de la información geográfica como un archivo de imagen digital conveniente para su visualización en una pantalla de ordenador.

⁴**Hardware:** Corresponde a todas las partes físicas y tangibles de una computadora.

⁵**Software:** Es todo el conjunto intangible de datos y programas de la computadora.

permitir poner a disposición del público catálogos de datos espaciales, que por medio de un simple navegador los usuarios pueden consultar, visualizar, utilizar y combinar la información geográfica según sus necesidades. Incorpora la interoperabilidad en forma de servicios de mapas en la web para poder integrar los datos y aumentar las funcionalidades (2).

Por las potencialidades y beneficios en términos de ahorro de recursos y en la toma de decisiones, los SIG constituyen una herramienta de soporte de gran utilización en el país. Cuba muestra los primeros avances de las IDEs con el surgimiento de la Infraestructura de Datos Espaciales de la República de Cuba (IDERC) el 16 de septiembre de 2005. Su objetivo principal es facilitar el acceso y la integración de la información geográfica que se gestiona, además de estandarizar los servicios y políticas que emplea para su realización (3).

La Universidad de las Ciencias Informáticas (UCI), cuenta con proyectos de desarrollo de software pertenecientes a diversos centros de desarrollo, que garantizan la informatización del país y aparejado el proceso de desarrollo de las tecnologías. El Centro de Desarrollo de *Software* GEySED se especializa en la creación de este tipo de sistemas informáticos a través de su Línea de Productos de *Software* (LPS) Aplicativos_SIG.

Los SIG desarrollados por el centro GEySED para la propia universidad donde radica, son empleados para apoyar la toma de decisiones de las diferentes esferas que coexisten, dependiendo de la localización de objetos, obtención y/o representación de información.

Actualmente la comunidad universitaria no aprovecha la mayoría de los SIG que han sido desarrollados, evidenciándose que no existen o hay pocos espacios colaborativos donde se discutan los temas de desarrollo, trabajos realizados y resultados obtenidos en esta materia. Como resultado significativo se demuestra la escasa divulgación y poca promoción de las potencialidades que ofrecen los sistemas.

La LPS Aplicativos_SIG cuenta con un número considerable de los mapas elaborados para los proyectos locales, nacionales y de otros países. Estos mapas no son utilizados por todos los usuarios interesados en mostrar en un momento determinado algún tipo de información geográfica en una de las aplicaciones que en la UCI se desarrollan, debido a que no poseen los permisos necesarios para realizar las acciones de consulta, búsqueda o consumo de la información. Entre las causas que dificultan la localización y consulta de los datos geográficos se presentan su baja disponibilidad y centralización.

Por lo anteriormente planteado el **problema a resolver** radica en ¿Cómo centralizar, integrar y proveer la información espacial de la Universidad de las Ciencias Informáticas?

Por tanto en la presente investigación el **objeto de estudio** lo representa el proceso de centralización, integración y provisión de la información geográfica.

El **campo de acción** lo compone la Infraestructura de Datos Espaciales en la Universidad de las Ciencias Informáticas.

Idea a defender: La Infraestructura de Datos Espaciales permitirá centralizar, integrar y proveer la información espacial de la Universidad de las Ciencias Informáticas.

Se ha definido como **objetivo general** desarrollar una Infraestructura de Datos Espaciales que permita centralizar, integrar y proveer la información espacial de la Universidad de las Ciencias Informáticas.

Como **objetivos específicos** se han identificado:

- ◆ Investigar las características de una Infraestructura de Datos Espaciales, así como sus objetivos y los beneficios que brindan su implementación.
- ◆ Determinar las tecnologías, herramientas y funcionalidades que se puedan integrar a una Infraestructura de Datos Espaciales en la UCI.
- ◆ Determinar los requisitos que debe tener la Infraestructura de Datos Espaciales para solucionar los problemas existentes en la UCI.

Para dar cumplimiento al objetivo planteado se han identificado las siguientes **tareas de la investigación:**

- ◆ Caracterización de las Infraestructuras de Datos Espaciales existentes a nivel internacional y nacional.
- ◆ Caracterización de las funcionalidades que soportan las IDEs identificadas.
- ◆ Caracterización de la información espacial que es manejada en la UCI, la manera de centralizarse, gestionarse y utilizarse.
- ◆ Fundamentación del uso de las tecnologías y herramientas a emplear para el desarrollo de un geoportal que soporte una IDE para la UCI.
- ◆ Identificación de las funcionalidades que debe brindar la IDE de la UCI.
- ◆ Diseño de la IDE de la UCI.
- ◆ Implementación de las funcionalidades de la IDE para la UCI.
- ◆ Validación y prueba del sistema.

El **muestreo intencional** es el procedimiento de muestreo seleccionado y se incluye dentro de los métodos **no probabilísticos**, el cual permite la selección de la información internacional y nacional como base para escoger la muestra ideal y más actualizada para su utilización a la hora de realizar el sistema. La población seleccionada para el análisis está compuesta por la Infraestructura de Datos de Espaciales de Europa (INSPIRE), Infraestructura de Datos Espaciales de España (IDEE), Instituto Colombiano de Datos Espaciales (ICDE), Instituto Nacional de Datos Espaciales de Brasil (INDE) y la Infraestructura de Datos Espaciales de la República de Cuba (IDERC).

De los **métodos teóricos** se selecciona el **análisis histórico – lógico** que permite la consulta de las bibliografías que ofrezcan información sobre el tema en cuestión, posibilitando caracterizar los componentes y funcionalidades de las IDEs para brindar una solución a la problemática existente.

De los **métodos empíricos** se emplea el **análisis documental** que permite realizar la búsqueda de la información referente al tema de investigación a través de diferentes fuentes bibliográficas.

Posibles resultados: Se pretende desarrollar una Infraestructura de Datos Espaciales técnicamente documentada en la Universidad de las Ciencias Informáticas disponible para todos los usuarios a través de un geoportal.

El presente trabajo está estructurado en cuatro capítulos:

Capítulo 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se presentan los conceptos necesarios para entender el contenido de la investigación. Se realiza un análisis profundo sobre el estado del arte⁶ a nivel nacional e internacional de las características de una IDE, así como los objetivos, estructura y principales funcionalidades que brindan. Se describe el uso de las herramientas y tecnologías en los que se apoya la investigación que posibilitan el avance de la solución.

Capítulo 2. CARACTERÍSTICAS DEL SISTEMA

Este capítulo se describe el dominio de la solución, estableciendo los requisitos funcionales y no funcionales. Se identifican los patrones de caso de uso a utilizar para la realización del Modelo de Caso de Uso del Sistema con el objetivo de una mayor comprensión de la aplicación.

Capítulo 3. DISEÑO E IMPLEMENTACIÓN

En este capítulo se realiza la descripción de la solución informática propuesta a partir del diseño del sistema. Se muestran las relaciones entre los componentes del sistema. Se definen los patrones

⁶ **Estado del arte:** Representa el avance y desarrollo que se tiene del tema en cuestión.

arquitectónicos y de diseño por los que se rige la aplicación y se realizan los artefactos que propone la metodología seleccionada durante el proceso de diseño e implementación.

Capítulo 4. VALIDACIÓN DEL SISTEMA

Se realizan un conjunto de diseños de Casos de Pruebas que permite al equipo de desarrollo verificar el correcto funcionamiento de los Casos de Usos implementados y que serán validados posteriormente para evaluar el comportamiento del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1. Introducción

El compartir información geográfica implica que sea posible adquirir, procesar, centralizar, distribuir, proveer y visualizar los datos geográficos disponibles, posibilitando que se fomente su publicación y uso. Para mejor comprensión del tema a desarrollar se abordan los principales conceptos de los contenidos a tratar. Se realiza un estudio de los siguientes elementos: Infraestructura de Datos Espaciales de Europa (INSPIRE), Infraestructura de Datos Espaciales de España (IDEE), Infraestructura Colombiana de Datos Espaciales (ICDE) e Infraestructura de Datos Espaciales de la República de Cuba (IDERC); además serán fundamentadas la selección de tecnologías y herramientas a emplear en el desarrollo de la aplicación.

2. Conceptos asociados al dominio del problema

2.1. Servidor de Mapa: Provee cartografías a través de la red tanto en modo vectorial como ráster.

Los servidores de mapas de código abierto permiten el desarrollo tanto interno como externo, no implican un coste económico por pago de licencia alguna, la mayoría son multiplataforma, soportan estándares OGC⁷ y diversas bases de datos. Se utilizan para desarrollar aplicaciones espaciales según los requisitos y el interés en contribuir a la mejora del servidor de mapas. Ejemplos de servidores de mapas de código abierto son: GeoServer, UMN, MapServer, Mapguide, Mapnik, Deegree (4).

2.2. Información geográfica o dato geográfico: Se denomina Información geográfica (IG) a los datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales. Dichos geodatos poseen una posición implícita (la población de una sección censal, una referencia catastral), o explícita (coordenadas obtenidas a partir de datos capturados mediante GPS⁸) (5).

2.3. Datos georreferenciados: Los datos georreferenciados poseen una orientación (entiéndase como coordenadas) que permite hacer uso de los mismos. Son los que forman el mapa base o mapa sobre el cual se referencian los datos temáticos (6).

2.4. Datos temáticos: Un conjunto de datos temáticos se refiere a los datos o capas que se encuentran relacionados con una tipo de información específica. Son los valores de las

⁷OGC (por sus siglas en inglés Open Geospatial Consortium): Consorcio Abierto Geoespacial

⁸GPS (Global Positioning System): Sistema de Posicionamiento Global es un sistema de navegación basado en satélites.

distintas capas de información geográfica, como por ejemplo: Clima, Edafología, Hidrología, Vegetación (6).

2.5. Metadatos: Para un colectivo de autores los metadatos pueden considerarse productos autónomos asociados a los conjuntos de datos geográficos que proporcionan información a los catálogos de datos, de servicios y las IDEs, de modo que se posibilite el uso compartido de los datos, además de organizar y mantener un inventario de datos o proporcionar información que posibilite la transferencia y el uso de estos (7).

2.6. Interoperabilidad: La Norma ISO⁹ 19119 establece que: “La interoperabilidad es la capacidad para comunicar, ejecutar programas o transferir datos entre varias unidades funcionales sin necesitar que el usuario tenga conocimiento de las características de esas unidades” (6).

2.7. Servicios: Acción y efecto de servir. Es un conjunto de actividades que buscan responder a las necesidades de un cliente (8).

2.7.1. Servicios web: Conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web. Permite la comunicación y el intercambio de datos entre aplicaciones y sistemas heterogéneos en entornos distribuidos que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones (9).

2.8. Sistemas de Información Geográfica (SIG): Es una tecnología de manejo de información geográfica formada por equipos electrónicos (*hardware*) programados adecuadamente (*software*) que permiten manejar una serie de datos espaciales (información geográfica) y realizar análisis complejos con estos siguiendo los criterios impuestos por el equipo científico (personal). Son por tanto cuatro los elementos constitutivos de un sistema de estas características: *hardware*, *software*, datos geográficos, equipo humano (10).

3. Infraestructura de Datos Espaciales (IDE)

Para realizar la conceptualización de la IDE de la UCI se toma como referencia la definición planteada por la Infraestructura de Datos Espaciales de España (IDEE). Se detallan además los servicios que integran una IDE y el análisis de varias infraestructuras mundiales que aporten elementos esenciales para el desarrollo de la aplicación.

IDEE considera que una “*IDE consiste básicamente en un sistema distribuido en la Red que permite a cualquier usuario mediante un simple navegador la búsqueda, localización, visualización, superposición en pantalla, consulta, análisis y, en ocasiones, descarga de los datos geográficos*

⁹ISO (Organization International of Standarization): Organización Internacional de Estándares.

disponibles en varios servidores que ofrecen mapas a varias escalas, nomenclátors, ortofotos, imágenes de satélite” (11).

3.1. Objetivos de la Infraestructura de Datos Espaciales

Los objetivos principales de la IDE son:

- ◆ Facilitar el acceso y la integración de la información espacial. Organizar y controlar los datos geográficos de un lugar, entidad o empresa. Integrar servicios de mapas y catálogos que brinden la información necesaria que apoye la seguridad en la toma de decisiones (12).
- ◆ Promover los metadatos estandarizados como método para documentar la información espacial. Estandarizar las interfaces de uso y la normalización de los servicios en que se basan (13).
- ◆ Permitir la utilización de manera fácil y eficaz de los datos geográficos existentes y de la información geográfica. Reutilizar la información geográfica generada en un proyecto para el desarrollo de nuevas aplicaciones, dado que ha sido un recurso de costosa producción y difícil acceso por varios motivos: formatos, modelos, políticas de distribución y falta de información.

3.2. Servicios utilizados en la Infraestructura de Datos Espaciales

La OGC tiene como principal objetivo definir estándares abiertos e interoperables dentro de los SIG y la World Wide Web (WWW). Estos ofrecen el correcto funcionamiento entre los sistemas de geoprocésamiento, facilitan el intercambio y uso de la información geográfica en beneficio de los usuarios. Dentro de las normas establecidas por la OGC se encuentra como formato de salida el Lenguaje de Mercado Geográfico (GML) y se definen los servicios: Servicio de Mapas en la Web (WMS), Servicio de Fenómenos en la Web (WFS), Servicio de Coberturas en Web (WCS), Servicio de Nomenclátor (Gazetteer) y Servicio de Catálogo Web (CSW) (14).

Es este conjunto de especificaciones se basan los servicios de las Infraestructuras de Datos Espaciales debido a que son estándares establecidos en el desarrollo de este tipo de aplicaciones, además aseguran la interoperabilidad con neutralidad tecnológica y han sido implementadas en numerosas herramientas de software.

3.2.1. Lenguaje de Mercado Geográfico (GML, Geography Markup Language por sus siglas en inglés): La OGC describe que “GML es una gramática XML¹⁰ para expresar las características geográficas. GML sirve como un lenguaje de modelado de sistemas

¹⁰**XML** (Extensible Markup Language, por sus siglas en inglés): Lenguaje de Mercado Extensible desarrollado por el World Wide Web Consortium (W3C). Se utiliza para el intercambio de datos estructurados. Más que un formato de archivos rígido, XML es un lenguaje que define los formatos aceptados que pueden utilizar los grupos para intercambiar información.

geográficos, así como un formato de intercambio abierto a las transacciones geográficas en Internet”. Como con la mayoría de las gramáticas basadas en XML hay dos partes de la gramática, el esquema que describe el documento y el documento de instancia que contiene los datos reales. Un documento GML se describe mediante un esquema de GML. Esto permite a los usuarios y desarrolladores describir conjuntos de datos geográficos genéricos que contengan puntos, líneas y polígonos (15).

3.2.2. Servicio de Mapas en la Web (WMS, Web Map Service por sus siglas en inglés): La OGC detalla que “*un WMS produce mapas de datos referenciados espacialmente de forma dinámica a partir de la información geográfica*”. Permite a los clientes la visualización de la superposición de imágenes de mapas desde varios WMS en una red de computadoras. Los mapas producidos por WMS son generalmente representados en un formato de imagen como PNG¹¹, GIF¹² o JPEG¹³ y ocasionalmente como vector basado en elementos gráficos en gráficos vectoriales escalables SVG¹⁴ o WebCGM¹⁵ (16).

3.2.3. Servicio de Fenómenos en la Web (WFS, Web Features Service por sus siglas en inglés): La OGC establece que “de manera similar al WMS, el WFS permite a los clientes recuperar y actualizar la información geoespacial codificada en Lenguaje de Mercado Geográfico (GML) a partir de múltiples WFS”. Entre los requisitos para la implementación de un WFS figuran que las interfaces deben estar definidas en XML; GML debe usarse para expresar características dentro de las interfaces” (17).

3.2.4. Servicio de Coberturas en Web (WCS, Web Coverage Service por sus siglas en inglés): La OGC define que “el servicio de cobertura Web (WCS) apoya el intercambio electrónico de datos geoespaciales como “coberturas”, es decir, información digital geoespacial que representan diferentes fenómenos en el espacio”. Proporciona acceso a los conjuntos potencialmente ricos y detallados de información geoespacial, en formas que son útiles para la prestación del lado del cliente, coberturas de varios valores, entrada en modelos científicos y otros clientes (18).

3.2.5. Servicio de Nomenclátor (Gazetteer en inglés): La OGC especifica que “es un servicio accesible desde la red que recupera la geometría conocida por uno o más características, teniendo en cuenta sus asociados conocidos con identificadores (nombre geográfico), que son especificados en tiempo de ejecución a través de una solicitud de consulta (filtro)”. Los

¹¹ **PNG** (por sus siglas en inglés, Portable Network Graphics)

¹² **GIF** (por sus siglas en inglés, CompuServe GIF)

¹³ **JPEG** (por sus siglas en inglés, Joint Photographic Experts Group)

¹⁴ **SVG** (por sus siglas en inglés, Scalable Vector Graphics)

¹⁵ **WCGM** (por sus siglas en inglés, Web Computer Graphics Metafile)

identificadores son palabras o términos que describen las características, que son bien conocidos por el Servicio de Nomenclátor, como un conjunto de nombres de un lugar y/o puntos de referencia. Cada instancia tiene un vocabulario asociado de los identificadores. Por lo tanto puede aplicarse a una región determinada, como por ejemplo un país, o algún otro grupo especializado” (19).

3.2.6. Servicio de Catálogo Web (CSW, Catalogue Service Web): La OGC particulariza que “el Servicio de Catálogo apoya la capacidad de publicar y buscar colecciones de información descriptiva (metadatos) para los datos. Los metadatos en los catálogos representan las características de los recursos que se pueden consultar y ser presentados para su evaluación y posterior procesamiento por los seres humanos y software”. Los CSW son necesarios para apoyar el descubrimiento y unión de los recursos de información registrados dentro de la comunidad de la información” (20).

3.3. Análisis de las soluciones existentes

Las Infraestructuras de Datos Espaciales existentes se rigen por normas y estándares establecidos que permiten una organización a nivel local, nacional e internacional. Con el objetivo de agrupar elementos significativos para el desarrollo de la IDE de la UCI se realiza el estudio y monitorización de varias infraestructuras funcionales en el mundo.

3.3.1. Infraestructura de Datos Espaciales de Europa (INSPIRE)

INSPIRE (Infrastructure for Spatial Information in Europe por sus siglas en inglés) es la IDE representativa de la Comisión Europea cuyo funcionamiento se recoge en la Directiva 2007/2/CE del Parlamento Europeo y del Consejo, de 14 de marzo de 2007, publicada en el Diario Oficial de la Unión Europea (DOUE) el 25 de Abril de 2007 (21).

Está orientada a las políticas comunitarias con un impacto directo o indirecto, sobre el medio ambiente y otras áreas. Su creación se basa en los estados miembros ya que brinda los objetivos, procedimientos administrativos nacionales y las leyes que los rigen que permiten mediante el uso de la información geográfica satisfacer sus necesidades prácticas. Proporciona el acceso a sus infraestructuras y enlaces a otros sitios a través del geoportal.

EL geoportal de INSPIRE ofrece los medios para la búsqueda de datos espaciales y servicios, tiene restricciones de acceso para diferenciar la entrada de los estados miembros de la Unión Europea. Gestiona un editor de metadatos permitiendo la creación de metadatos acorde a las reglas establecidas. Posee un validador de metadatos cuyo propósito es chequear que estos cumplan con lo establecido por la regulación de la directiva, un visor de la información

geográfica de Europa que brinda la posibilidad de búsqueda, localización y la variabilidad de la escala del mapa para ofrecer más información sobre los datos geográficos. Incluye además conferencias y librerías que aportan más información sobre la IDE (21).

3.3.2. Infraestructura de Datos Espaciales de España (IDEE)

La IDEE es un proyecto coordinado por el Consejo Superior Geográfico (CSG) y la Dirección General del Instituto Geográfico Nacional que asume la Secretaría Técnica del CSG. La IDEE constituye un sistema distribuido de servidores accesibles mediante la red. Actúa como un nodo nacional de la INSPIRE y contiene el registro de proveedores de datos y servicios a nivel español, además de participar en los grupos de trabajo que elaboren las directivas europeas y el desarrollo técnico europeo. Permite la integración de las IDEs regionales (ej. Infraestructura de Datos Espaciales de Catalunya (IDEC)) para facilitar el listado de los proveedores de datos y servicios a su nivel (12).

La IDEE contiene un geoportal con una interfaz que incluye siete idiomas y ofrece acceso a otros geoportales, datos y servicios de la administración nacional y regional. Permite el enlace y referencia a IDEs de otros países perteneciente al continente europeo como Portugal y Francia. Contiene un visor con funcionalidades de navegación y un control de capas, posibilita la gestión de los servicios de mapas en la web. Permite la búsqueda y localización de objetos en el espacio, la búsqueda por nombre, el análisis territorial, generación de metadatos, añadir y listar servicios. Provee el acceso a fenómenos y centro de descarga. Contiene un catálogo de datos y un catálogo de servicios. Establece enlaces hacia páginas de interés como son la Organización Internacional de Estándares (ISO), INSPIRE, OGC, GeoSur, Google Earth y otras. Brinda videos y herramientas gratuitas, revistas electrónicas, ejemplos de API¹⁶ y un rincón del desarrollador para estudiar e indagar más sobre las potencialidades de esta IDE (11).

3.3.3. Instituto Colombiano de Datos Espaciales (ICDE)

El ICDE define el conjunto de políticas, organizaciones, estándares y tecnologías que trabajan conjuntamente para producir, compartir y usar información geográfica necesaria para soportar el desarrollo de este país. Esta infraestructura permite acceder mediante redes distribuidas a información espacial de diversas fuentes ubicadas en diferentes entidades. Contiene un geoportal que permite el enlace al geoportal del Telecentro Regional y al geoportal Política de Tierra, a sitios como el Instituto Colombiano de Geología y Minería, Instituto Geográfico Agustín Codazzi, Instituto de Meteorología, Hidrología y Medio Ambiente, entre otros que trabajan con

¹⁶ API (Application Program Interface): Interfaz de Programación de Aplicaciones.

datos geoespaciales. Contiene varios visores que permiten tener acceso a todas las funcionalidades de visualización de información geográfica como acercar, alejar, paneo, activar y desactivar capas, consultar datos geoespaciales. Agrega más capas al visor provenientes de otros servicios de visualización que cumplan con la especificación del WMS para aumentar el nivel de información, entre ellos se encuentra el visor de rutas, visor de líneas de transmisión electrónicas, visor en 3D, visor de ecosistemas continentales, marino y costero. Permite listar servicios y consultar metadatos.

3.3.4. Instituto Nacional de Datos Espaciales de Brasil (INDE)

El INDE permite la disponibilidad, el intercambio y acceso a datos, información geoespacial y servicios relacionados a través de una red integrada de servidores de Internet. El Directorio de Datos Espaciales de Brasil (DBDG) posibilita reunir a productores, gestores y usuarios de información geoespacial en el ciberespacio. El geoportal brasileño permite que los usuarios accedan a los recursos distribuidos del DBDG. Establece normas y patrones de los datos geográficos, brinda un visor que posibilita la búsqueda y localización de los datos geográficos del país, permite la consulta y acceso a los metadatos, contiene un catálogo de metadatos y un catálogo de servicios y posibilita además el enlace hacia otros sitios nacionales e internacionales referentes al tema.

3.3.5. Infraestructura de Datos Espaciales de la República de Cuba (IDERC)

La IDERC se lanzó el 30 de noviembre de 2004. Abarca las políticas, tecnologías y estándares necesarios que permiten la administración, acceso y utilización de los datos espaciales a nivel nacional en función de la toma de decisiones económicas, políticas y sociales (22).

El geoportal es el punto de acceso nacional a la información geográfica que contiene un visor del mapa topográfico de la República de Cuba. Permite la visualización de las capas del mapa y brinda las funcionalidades de acercar, alejar, buscar y localizar datos geográficos según la provincia y municipio, la búsqueda de servidores de mapas, información geográfica, imágenes de satélites, direcciones y otros tipos de recursos. Brinda servicios Nomenclátor y un diccionario de nombre geográficos accesible mediante mapas interactivos y los Mapas de Población del Anuario Estadístico de Cuba editado en 2003, además del catálogo de metadatos. Posibilita el enlace hacia los portales provinciales como es el caso de Villa Clara, Matanzas y hacia infraestructuras internacionales como son: Europa, Australia, Canadá, Chile, Colombia, República Dominicana, Francia, Italia, Nicaragua y Portugal (22).

La IDERC establece dentro de sus normas la ISO/TC 211, esta tiene como propósito la promoción de los estándares internacionales dentro de la comunidad hispanohablante especializada de manera consistente con el mandato del Instituto Panamericano de Geografía e Historia. Dentro de sus aspectos principales se incluyen la formulación técnica de normas, organizaciones que formulan normas geográficas o relacionadas con la geografía, prioridades de las normas, pruebas de interoperabilidad, así como la rapidez para formular y aprobar especificaciones técnicas.

4. Propuesta de solución

Luego del estudio realizado a las Infraestructuras de Datos Espaciales caracterizadas anteriormente, se define la creación de una IDE para la UCI que centralice y controle toda la información geográfica y servicios que se gestionan en la universidad, que permita integrar catálogos de datos y servicios que apoyen la toma de decisiones y que incluya un geoportal como punto de acceso fundamental a la información geográfica.

De manera que la centralización de toda la información geográfica de la UCI en un punto de acceso común, debe posibilitar la reutilización de los datos para el desarrollo de nuevas aplicaciones, debe contener recursos que pondrán a disposición del público los datos geográficos y servicios, además de añadir un visor que permita la búsqueda, visualización, manejo y combinación de los datos geográficos. La IDE para la UCI gestionará los servicios: Servicio de Mapas en la Web, el Servicio de Fenómenos en la Web y un Catálogo de servicios. Para un mayor intercambio y gestión de la información poseerá vínculos con otros portales nacionales e internacionales.

5. Tecnologías utilizadas en el desarrollo del sistema

5.1. Metodología de desarrollo de software: Proceso Unificado de Desarrollo (RUP, Rational Unified Process por sus siglas en inglés)

Proceso Unificado de Desarrollo es un proceso de desarrollo de software que se adapta según las necesidades de los desarrolladores y del cliente, debido a que la comunicación entre estos debe ser fluida, permitiendo llegar a un acuerdo para evitar desavenencias futuras, además integra el Lenguaje Unificado de Modelado (UML) y está basado en componentes interconectados a través de una interfaz bien definida. RUP se caracteriza por ser dirigido por Casos de Uso¹⁷, en lo adelante CU, centrado en la arquitectura, iterativo e incremental (23).

¹⁷**Casos de Uso** (CU): fragmento de funcionalidades del sistema que proporciona al usuario un resultado importante, representan los requisitos funcionales.

Es dirigido por CU debido a que el desarrollo del software contiene un hilo conductor que avanza según el flujo de trabajo que se genera a raíz de la creación de los CU, estos permiten especificar los requisitos funcionales del sistema y constituye una guía para el diseño, implementación y prueba del mismo.

El proceso de desarrollo de software es centrado en la arquitectura ya que muestra una vista del diseño completo de las características más importantes del sistema. Se basa en la comprensión, la capacidad de adaptación al cambio y la reutilización. La arquitectura debe diseñarse para que el sistema funcione adecuadamente según los CU establecidos, de ahí que deben evolucionar simultáneamente y a medida que los CU se especifican y maduran se descubre más de la arquitectura (23).

RUP propone un proceso de desarrollo interactivo e incremental. De manera que el trabajo se divida en subproyectos o en partes más pequeñas donde cada una de estas representa una iteración que propicia un incremento en el software. Cada iteración debe ser controlada para disminuir el riesgo de un incremento, para que el software sea concluido en el tiempo planificado, se trabaje con eficiencia donde el esfuerzo tenga un resultado a corto plazo y no indefinidamente.

El ciclo de RUP está compuesto por cuatro fases: Inicio, Elaboración, Construcción y Transición y cada una de ellas se divide en iteraciones, a su vez cada iteración trabaja en un número de disciplinas, entre ellas se encuentran: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas (24).

Se selecciona esta metodología debido a que brinda una guía para ordenar las actividades de un equipo y dirige las tareas de cada desarrollador por separado y del equipo como un todo. Especifica los artefactos que deben desarrollarse, ofrece criterios para el control y medición de los productos. El equipo de desarrollo es cambiante y es necesario que todo el trabajo quede debidamente documentado, para que no existan en un futuro problemas en la creación de una nueva versión o actualización sistema.

5.2. Lenguaje de modelado: Lenguaje de Modelado Unificado (UML, Unified Modeling Language por sus siglas en inglés)

El Lenguaje de Modelado Unificado, en lo adelante UML, es el lenguaje más conocido y utilizado en el modelado de sistemas de software. Es un lenguaje gráfico que permite visualizar, especificar, construir, describir métodos y documentar el sistema a desarrollar que cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. Ofrece un estándar para detallar un modelo, incluyendo características conceptuales como procesos de

negocio y las funciones y aspectos concretos como expresiones de lenguaje de programación, esquemas de bases de datos y componentes reutilizables. Se puede emplear en el desarrollo de un software como soporte a una metodología de desarrollo de software como RUP, pero no se especifica cuál metodología o proceso se debe usar (25).

Se selecciona este lenguaje de modelado ya que está estrechamente relacionado con la metodología empleada. Además, en el proyecto donde radican los desarrolladores de la aplicación existe personal capacitado que puede brindar conocimientos para esclarecer las dudas existentes porque cuentan con experiencia en el uso de este lenguaje en la creación de aplicaciones anteriores.

5.3. Leguajes de Programación

5.3.1. Hipertexto Pre-procesador (PHP, Hypertext Pre-processor por sus siglas en inglés)

PHP es un lenguaje de programación interpretado, multiplataforma, diseñado originalmente para la creación de páginas web dinámicas con acceso a información almacenada en una base de datos. Se usa principalmente para la interpretación del lado del servidor donde el código fuente escrito en PHP, es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. PHP posibilita el uso de la técnica de programación orientada a objetos, tiene manejo de excepciones y puede interactuar con cualquier servidor web. Establece conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL¹⁸ y PostgreSQL¹⁹. Es libre por lo que se presenta como una alternativa de fácil acceso para todos (26).

Se selecciona el lenguaje de programación debido a su característica de ser multiplataforma lo que viabiliza el ser ejecutado en cualquier servidor, facilita la comunicación con bases de datos y además el equipo de desarrollo tiene conocimientos sólidos de este lenguaje. La herramienta para la confección del geoportal es el CMS Drupal que tiene definido este lenguaje como base para su funcionamiento.

5.3.2. Lenguaje de Marcado de Hipertexto (HTML, HyperText Markup Language por sus siglas en inglés)

¹⁸**MySQL:** es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, bajo el esquema de licenciamiento dual, GNU y GPL.

¹⁹**PostgreSQL:** es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

HTML es el lenguaje marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Utiliza etiquetas o marcas, que son breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, imágenes y demás elementos, en la pantalla del ordenador. Los elementos son la estructura básica de HTML, estos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML (27).

Se selecciona este lenguaje para el diseño de la aplicación por su fácil comprensión, su simplicidad al trabajar y la posibilidad de observar los resultados obtenidos mientras se está en el proceso de desarrollo y construcción.

5.3.3. JavaScript

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objeto, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y página web dinámicas, en bases de datos locales al navegador (28).

En una aplicación cliente, las sentencias JavaScript embebidas en un documento HTML pueden reconocer y responder a eventos generados por el usuario. JavaScript trabaja en base a un simple sistema en tiempo de ejecución, el cual permite el uso de tipos de datos básicos como enteros, números de punto flotante, booleanos y strings (28).

Se hará uso de JavaScript debido a que soporta un sistema simple de objetos basado en instancias que provee capacidades significativas. Además soporta funciones sin ningún requerimiento especial al declararlas, pueden ser propiedades de objetos y se ejecutan como métodos bastante flexibles.

5.4. Arquitectura: Cliente-Servidor

Una arquitectura es un conjunto de reglas, definiciones, términos y modelos que se emplean para producir un producto.

La arquitectura Cliente/Servidor agrupa conjuntos de elementos que efectúan procesos distribuidos y cómputo cooperativo. En la arquitectura Cliente/Servidor el cliente es la entidad que realiza solicitudes o peticiones y el servidor quién recibe la solicitud, la procesa y retorna un resultado. De esta manera el cliente interactúa con los usuarios finales mediante una interfaz gráfica de

usuario y el servidor acepta conexiones desde un gran número de clientes y brinda respuestas. El cliente y el servidor no tienen que estar precisamente localizados en un mismo lugar para que pueda funcionar el sistema, aunque pueden coincidir (29).

Esta arquitectura posibilita mayor escalabilidad y proporciona un reparto adecuado de las tareas para dichos sistemas. Además no se utiliza la metodología de compartición de archivos, ya que todos los accesos a la información se llevan a cabo a través de peticiones por medio de comunicación y tiene gran flexibilidad de establecer sesiones con múltiples servidores y manejo de información en varias bases de datos (29).

Se selecciona esta arquitectura porque posibilita que el usuario interactúe con el sistema sin que este conozca lo que sucede a nivel de aplicación. Permite que el servidor tenga control sobre los accesos, recursos y la integridad de los datos de forma que un programa cliente defectuoso o no autorizado no pueda causar daño. Facilita la actualización de datos y recursos, cualquier elemento puede ser incrementado o mejorado en cualquier momento o se pueden añadir nuevos nodos a la red (clientes y/o servidores). Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes es posible reemplazar, reparar, actualizar o incluso trasladar un servidor, mientras que sus clientes no se vean afectados por ese cambio o se afecten mínimamente.

6. Herramientas a utilizar

6.1. Librería de visualización de mapas: OpenLayers

OpenLayers es una biblioteca que implementa una API JavaScript del lado del cliente para la creación de aplicaciones geográficas. Permite la visualización de mapas y marcadores cargados desde diferentes fuentes y resulta fácil para mostrar un mapa dinámico sobre cualquier página web. Su desarrollo está sujeto al fortalecimiento del uso de la información geográfica de todo tipo. Constituye una herramienta totalmente gratuita, desarrollada bajo código abierto y publicado bajo la licencia BSD²⁰. OpenLayers explota métodos estándar que son definidos por las organizaciones rectoras sobre el uso de la información geográfica.

Proporciona funciones que permiten al usuario interactuar con el mapa, entre ellas se encuentran:

- ◆ LayerSwitcher: Control para la gestión de la visibilidad de capas.

²⁰**Licencia BSD (Berkeley Software Distribution por sus siglas en inglés):** Es una licencia de software libre permisiva como la licencia OpenSSL o la MIT License. Permite el uso del código fuente en software no libre.

- ◆ PanZoomBar: Crea una barra de zoom y un panel de navegación que permite realizar las operaciones de ZoomIn y ZoomOut sobre el mapa.
- ◆ OverViewMap: Crea un pequeño mapa de navegación y muestra la posición actual del mapa principal, constituyendo así otra herramienta de navegación. El rectángulo que se muestra puede ser movido para cambiar la posición actual del mapa principal.
- ◆ MouseToolBar: Barra de herramientas que permite escoger como utilizar el ratón entre navegación y operaciones de zoom.
- ◆ ScaleLine: Muestra una escala gráfica sobre el mapa.
- ◆ MousePosition: Muestra las coordenadas actuales del cursor sobre el mapa (30).

Se selecciona OpenLayers porque facilita la visualización de mapas dinámicos en el visor que incluirá el geoportal, hace gran uso de métodos definidos sobre la información geográfica por organizaciones internacionales y realiza la personalización de funciones que proveen al usuario un contacto interactivo con la aplicación que se crea. La velocidad de procesamiento de la información geográfica de OpenLayers en comparación con otras librerías de visualización es superior, es una característica que definió la inclusión de esta herramienta en el proceso de desarrollo del software.

6.2. Librería para interfaz de usuario: jQuery

jQuery es una librería JavaScript que permite añadir multitud de efectos y funcionalidades a las páginas web. Funciona como un framework y es accesible a la mayoría de navegadores actuales. Esta librería permite abrir imágenes en una ventana dentro de la propia página, crear galerías de imágenes, validar un formulario en el servidor usando AJAX, gestionar las cookies de los usuarios, crear menús dinámicos, entre otras operaciones (31).

La gran ventaja de la función de jQuery, es que la página se puede manipular en cuanto se ha cargado su código HTML mientras que la función de JavaScript espera a que se carguen todos los elementos de la página, incluyendo todas las imágenes. De esta forma, las aplicaciones realizadas con jQuery pueden responder de forma mucho más rápida que las aplicaciones JavaScript tradicionales.

jQuery UI es una biblioteca de componentes para el framework jQuery que provee un conjunto de plugins, widgets y efectos visuales. Cada componente o módulo se desarrolla de acuerdo a la filosofía de jQuery. Dentro de los componentes de interacción de jQuery UI, cuenta con métodos de funciones predeterminadas para añadir comportamientos a los nodos obtenidos o creados. Brinda efectos visuales para la duración en la aplicación a modo de transición entre estados y la posibilidad de invocar a un método de callback después de la transición.

Se selecciona esta librería para realizar el estilo de la página inicial del visor, este incluye la estructura de los componentes de control y navegación que se mostrarán. Se utiliza la biblioteca jQuery UI ya que contiene las funciones básicas para implementar en el módulo nombrado "ide_visor" y añade comportamientos complejos a los elementos de este, facilitando la reutilización de funciones y eventos que simplifican el desarrollo del lado del cliente. Es una biblioteca pequeña, rápida, ligera y además compatible con todos los navegadores.

6.3. Servidor de mapas: MapServer

Los servidores de mapas permiten la interacción con la información espacial almacenada en servidores de datos espaciales accesibles a través de la web. MapServer es un entorno de código abierto para la publicación de datos espaciales y aplicaciones de cartografía interactiva para la web. Es liberado bajo la licencia estilo MIT²¹ y funciona en las principales plataformas: Windows, Linux, Mac OS X. Ofrece soporte a los lenguajes de scripting más populares: PHP MapScript, Python MapScript, Perl MapScript, Ruby MapScript, MapScript Java y .NET MapScript. Debido a que se considera una aplicación ejecutable CGI²², es capaz de admitir formatos de datos tanto vectoriales (ESRI shapfiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL, a través de OGR²³) como ráster (TIFF/GeoTIFF, EPPL7, a través de GDAL²⁴). Ofrece ventajas como la posibilidad de presentar los datos a otros usuarios empleando Internet, es uno de los pocos productos que ofrecen soluciones a los proyectos que disponen de mapas en múltiples formatos (GDAL/OGR), acceso a datos y mantenimiento (GDAL/OGR, PostGIS²⁵) (32).

Por lo expuesto anteriormente y por las características especiales del trabajo en cuestión, se especifica MapServer como servidor de mapas. Permite la publicación de datos espaciales y cartografías para su uso en la web. Además es compatible con la mayoría de las plataformas existentes, incluye soporte para PHP que es el lenguaje de programación a utilizar. Admite formatos de datos que son generados en la UCI (json, shape y xml) y especificaciones establecidas por la OGC necesarias para desarrollo de la IDE.

²¹**Licencia MIT:** El texto de la licencia no tiene copyright, lo que permite su modificación. La licencia MIT es muy parecida a la licencia BSD en cuanto a efectos.

²²**CGI (Common Gateway Interface):** CGI es un programa de interfaz que permite utilizar programas externos para realizar una función específica.

²³**OGR:** Es una librería Open Source para el procesamiento de datos vectoriales.

²⁴**GDAL:** Una librería para formatos de datos geoespaciales ráster.

²⁵**PostGIS:** Extiende PostgreSQL, permitiendo almacenar y manipular tipos de datos geográficos.

6.4. Entorno integrado de desarrollo: Netbeans 7.0

Es un entorno de desarrollo integrado libre. Soporta lenguajes como PHP y C con C++. Más que un entorno de desarrollo, Netbeans es una comunidad en crecimiento compuesta por cientos de socios en el mundo. Existe un gran número de módulos que permiten su extensión y las aplicaciones creadas con la plataforma pueden ser construidas a partir de módulos, los cuales pueden ser añadidos para extender las funcionalidades de las aplicaciones.

Entre las características más relevantes, resalta la detección y reparación de errores, capacidad de realizar sugerencias de cambios y reportar al usuario de posibles pérdidas de información. Incluye un depurador y un selector de vistas que posibilita navegar por las líneas de código de un programa, siendo esto la base de la detección de errores en la lógica del programa.

Se hará uso de Netbeans como entorno de desarrollo porque está diseñado para funcionar fácilmente. Además posee administración de las interfaces y configuración del usuario, administración de almacenamiento y administración de ventanas. Constituye un sistema multiplataforma e inserta soporte para APIs y plugins que permitan un mayor desarrollo. Es libre, robusto e intuitivo.

6.5. Herramienta de modelado de software: Visual Paradigm

Es una herramienta CASE²⁶. Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad para el analista. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Visual Paradigm se caracteriza por ser un sistema multiplataforma, distribuido bajo una licencia comercial y gratuita. Su diseño se centra en los casos de usos y enfocado al negocio que generan un software de mayor calidad además de ser capaz de ofrecer ingeniería directa e inversa (código a modelo, código a diagrama) (33).

Se selecciona Visual Paradigm como herramienta de modelado pues es un sistema liberado bajo licencia comercial y gratuita, ofrece ventajas para el desarrollo del software como la planificación,

²⁶CASE (por sus siglas en inglés Computer Aided Software Engineering): Ingeniería de Software Asistida por Computadora.

que mantendrá al equipo de trabajo actualizado en cada fase del proceso. Permite la creación de los artefactos que se modelan haciendo uso de la metodología seleccionada.

6.6. Sistema de Gestor de Contenido: Drupal

Drupal es un sistema gestor de contenidos (CMS²⁷) de extensa configuración. La aplicación está basada en código abierto liberada bajo la licencia GNU/GPL y se encuentra escrito en PHP. Cumple con los estándares establecidos para la creación de sitios web, y permite la creación de diferentes estilos de sitios web. El sistema está caracterizado por admitir la publicación de artículos, galería de imágenes, servicios como foros, encuestas, administración de usuarios, permisos y blogs. Se basa en una arquitectura modular, ya que contiene módulos predefinidos y es extensible pues posibilita la incorporación de otros que incrementen las funcionalidades sin tener que afectar a los ya creados. Los módulos son divididos en 3 grupos, núcleo: son los módulos que se instalan con Drupal por defecto, contribuidos: son los módulos compartidos por la comunidad de Drupal y liberados bajo la licencia GNU/GPL y personalizada: son los módulos creados por los propios desarrolladores con fines específicos (34).

Dentro de las principales características que ofrece Drupal destaca un robusto sistema de ayuda online y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores. El código fuente está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario de otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades. Posee un entorno de personalización que está implementado en el núcleo. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario. Los administradores de este CMS no tienen que establecer permisos para cada usuario, en lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por esa categoría.

El contenido creado en Drupal es funcionalmente un objeto (nodo) que permite un tratamiento uniforme de la información. Posibilita independencia de la base de datos debido que incorpora una capa de abstracción que permite incorporar fácilmente soporte para otras bases de datos. Es un sistema multiplataforma. Por otro lado, al estar implementado en PHP, es totalmente portable. El sistema está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Posee un sistema de caché que elimina consultas de la base de datos, incrementando

²⁷**CMS** (Content Management System): Un sistema de gestión de contenidos es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web.

el rendimiento y reduciendo la carga del servidor. Sus principales beneficios son: modularidad, escalabilidad, código abierto, multiplataforma, multi-idioomas, fácil de administrar, extensible, programable y personalizable (35).

Se selecciona Drupal como gestor de contenidos debido a que todo el contenido es totalmente indexado en tiempo real y se puede consultar en cualquier momento, cuenta con módulos básicos para su uso incluyendo los que son añadidos, y posibilita la personalización y creación de módulos. Por otra parte contiene un sistema de temas que separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio o aplicación web. Se pueden crear plantillas con HTML y/o con PHP y agregarla al sistema. Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación. Se realiza el análisis y administración, seguimiento y estadísticas, registros e informes vía web.

7. Conclusiones parciales del capítulo

Los conocimientos y elementos obtenidos luego del análisis a varias IDEs funcionales mundialmente, sirven como base teórica en el desarrollo de una IDE para la UCI. La centralización de la información geográfica de la UCI a través de este sistema, permite un mejor proceso de gestión, provisión y control de la misma. El uso de las tecnologías y herramientas seleccionadas para el desarrollo del sistema posibilita un costo monetario nulo por concepto de pago de licencias. Debido a que la creación de una IDE es un proceso que reúne aspectos como la organización que integra normas, políticas, personal y una estructura, el grupo de desarrollo se restringe a la parte técnica de la aplicación, por tanto cuando se realice la entrega del producto a la UCI, esta última será la encargada de gestionar todos los elementos organizacionales.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

1. Introducción

IDEUCI cuenta con un conjunto de funciones necesarias para la centralización, acceso y visualización de la información geográfica. Los usuarios del sistema serán los responsables del manejo y control de la información, accediendo sólo a las funcionalidades del sistema a las que tienen permisos asignados según su rol. En el presente capítulo se definen las características tanto cuantitativas como cualitativas que debe proporcionar el sistema, las cuales son descritas en términos de requisitos funcionales y no funcionales.

2. Modelo de Dominio

El Modelo de Dominio es una representación visual estática del objeto real del proyecto. Es decir, un diagrama con los objetos que están relacionados con el proyecto que se hará y las relaciones que hay entre ellos. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema. El modelo de dominio ayuda a los usuarios, desarrolladores y otros interesados en utilizar un vocabulario común, de manera que la terminología común sea necesaria para compartir el conocimiento (36).

En general el modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Imitar en alguna medida la realidad del funcionamiento interno del software. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. El modelo de dominio es utilizado por el analista como un medio para comprender el sector al cual el sistema va a servir.

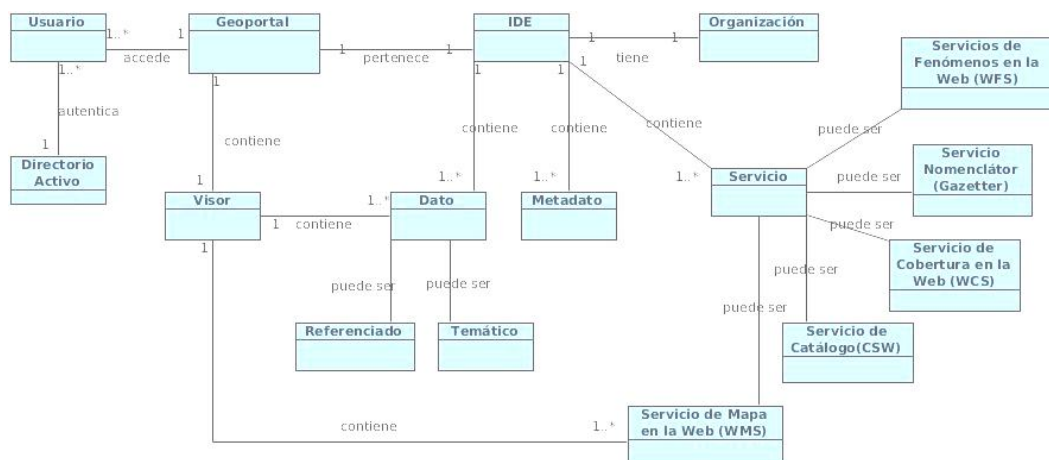


Figura 1: Modelo de Dominio

2.1. Descripción de las clases del dominio

IDE: Sistema estandarizado integrado por un conjunto de recursos informáticos cuyo fin es visualizar y gestionar información geográfica disponible en la red de la UCI.

Geoportal: Sitio web cuyo objetivo es ofrecer al usuario el acceso a una serie de recursos y servicios basados en información geográfica, entre los que suelen encontrarse buscadores de información geográfica, documentos, aplicaciones, servicios OGC.

Usuario: Persona que accede y hace uso de la aplicación.

Directorio Activo: Es el centro que contiene la base de datos de los usuarios que trabajarán en el sistema, además de la cartografía por la que estará compuesta el visor de la aplicación.

Visor: Muestra información geográfica generada a partir de diferentes fuentes de datos.

Dato: Información concreta sobre hechos, elementos, que permite estudiarlos, analizarlos y utilizarlos.

Dato Referenciado: Son los que forman el mapa base o mapa sobre el que se referencian los datos temáticos.

Dato Temático: Datos o capas que se encuentran relacionados con un tipo de información específica.

Metadato: Permiten que el usuario conozca las características de los datos de manera que se facilite la búsqueda y selección de los datos que le interesan al usuario y que sea capaz de explotarlos de la manera más eficaz posible.

Servicio: Utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Servicio de Mapas en la Web (WMS): Genera mapas de datos referenciados espacialmente de forma dinámica a partir de la información geográfica.

Servicio de Fenómeno en la Web (WFS): Permite acceder a los datos y disponer de la información vectorial que ofrecen los mismos.

Servicio de Catálogo (CSW): Permite localizar la información geográfica que se necesita en base a los metadatos que la definen.

Servicio de Cobertura en la Web (WCS): Similar al WFS pero para el tipo de dato ráster, como son imágenes satelitales y modelos digitales del terreno.

Servicio de Nomenclátor (Gazetter): Permite localizar fenómenos geográficos referenciados por nombre. El servicio une cada nombre geográfico con su localización en base a coordenadas.

Organización: Incluye el personal humano, estructura organizativa, estándares y normas que hacen que los sistemas puedan interoperar.

3. Requisitos Funcionales (RF)

La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requisitos conforme estos se transforman en un sistema operacional. Por su parte los requisitos no funcionales serán aquellos que determinarán las restricciones en el diseño, las propiedades o cualidades que debe tener la aplicación para que sea un producto atractivo, usable, rápido o confiable (37).

Como resultado del análisis realizado a las IDEs estudiadas de acuerdo con las principales funcionalidades que implementan, se definieron los siguientes requisitos funcionales que compensarán las necesidades que existen en la universidad:

La aplicación cuenta con 6 módulos:

Módulo de Autenticación

RF 1. Autenticar Usuario

RF 2. Cerrar Sesión

Módulo de Navegación

RF 3. Acercar determinada región del mapa

RF 4. Alejar determinada región del mapa

RF 5. Visualizar todo el mapa

RF 6. Modificar el centro del mapa

RF 7. Visualizar diferentes regiones del mapa

Módulo de Visualización

RF 8. Habilitar capas del mapa

RF 9. Deshabilitar capas del mapa

RF 10. Ver datos de las capas

Módulo de Consulta

RF 11. Realizar Información puntual

RF 12. Listar Servicios Existentes

RF 13. Añadir Servicio Listado

Módulo de Gestión de Servicios

RF 14. Adicionar Servicio de Mapas en la Web

RF 15. Modificar Servicio de Mapas en la Web

RF 16. Buscar Servicio de Mapas en la Web

RF 17. Eliminar Servicio de Mapas en la Web

RF 18. Adicionar Servicio de Fenómenos en la Web

RF 19. Modificar Servicio de Fenómenos en la Web

RF 20. Buscar Servicio de Fenómenos en la Web

RF 21. Eliminar Servicio de Fenómenos en la Web

RF 22. Buscar Servicio.

RF 23. Modificar Servicio de Usuario

RF 24. Buscar Servicio de Usuario

RF 25. Eliminar Servicio de Usuario

Módulo de Impresión

RF 26. Exportar Mapa

4. Requisitos No Funcionales (RNF)

La aplicación debe permitir la confiabilidad, integridad y disponibilidad de la información, además debe tener una interfaz agradable posibilitando una interacción de fácil comprensión y manipulación al usuario. Para ello se han establecido los siguientes requisitos no funcionales:

4.1. Usabilidad

El sistema puede ser utilizado por aquellas personas que tengan conocimientos básicos de computación y en el manejo de información geográfica. El diseño es sencillo sin mucha complejidad, de manera que permita la interacción dinámica y descifrable del sistema.

4.2. Fiabilidad y disponibilidad

La información manejada por el sistema está protegida de acceso no autorizado y divulgación. La información y las funcionalidades del sistema están disponibles las 24 horas de los 7 días de la semana, permitiéndole al usuario acceder a ellas. El sistema realizará una adecuada gestión de los errores que puedan ocurrir, garantizando la estabilidad e integridad del mismo. Ante una falla del sistema, el mismo pasa a estado de mantenimiento e indica al usuario las posibles causas del error, en caso de que el sistema no pueda reconocerlo se le informa a los usuarios los datos del administrador para su contacto.

4.3. Eficiencia

El tiempo de respuesta estará dado por la cantidad de información a procesar, entre mayor cantidad de información mayor será el tiempo de procesamiento. Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

4.4. Soporte

El sistema recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y el Centro GEySED, el cual no debe exceder de 3 meses.

4.5. Apariencia o interfaz externa

El sistema debe contar con una apariencia profesional e interfaces de usuario sencillas, diseñadas de la manera más atractiva posible. Los colores que deben predominar en el sistema deben tener un tono claro, evitando el uso de colores fuertes que perjudiquen la visión de los usuarios. Se utilizará para los textos el tipo de fuente Arial con tamaño mínimo 10 y máximo 14, el color de fondo será claro y utilizará íconos para la interacción con el usuario de manera intuitiva.

4.6. Hardware

Clientes

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

- Se requiere tarjeta de red

Servidores

- Procesador 3 GHz como mínimo.
- El Servidor de BD tenga como mínimo 2GB de RAM y 40 GB de disco duro.
- El Servidor de Mapas tenga como mínimo 2GB de RAM y 40 GB de disco duro
- Se requiere tarjeta de red.

4.7. Software

Clientes

- Un navegador como Mozilla Firefox, Zafari, cualquiera que cumpla con los estándares W3C.
- Cualquier sistema operativo que permita la ejecución de un navegador web.

Servidores

- Sistemas operativos GNU/Linux Ubuntu Server 11.04.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte para datos espaciales.
- MapServer 5.6.6 o superior, con extensión PHP mapscript.

4.8. Seguridad

La aplicación contará con asignación de roles por parte de los administradores del sistema que establecerán los permisos de escritura, lectura y modificación para cada uno de los usuarios, según las responsabilidades que cumplan. Por otro lado el sistema contará con una revisión continua para saber el estado en que se encuentra.

5. Modelo de Casos de Uso del sistema

Luego de tener una visión del funcionamiento del sistema mediante el modelo de dominio y haber establecido los requisitos, para hacer más eficaz el tratamiento de estos se agrupan en Casos de Usos (CU) que no es más que una descripción de un proceso de principio a fin relativamente amplia, esta suele abarcar muchos pasos, eventos, acciones y transacciones que se requieren para el correcto

funcionamiento del sistema. Para que los CU fueran abarcadores y específicos se utilizaron Patrones de CU que permiten describir lo que el sistema debe hacer y cómo este interactúa con los usuarios; son utilizados generalmente como plantillas que describen como deberían ser estructurados, organizados y modelados los CU.

Dentro de estos patrones se utilizaron se encuentra el Patrón CRUD²⁸, que consiste en un caso de uso para administrar la información que permite modelar las diferentes operaciones para administrar una entidad, tales como crear, leer, cambiar y eliminar. Además del Patrón Múltiples Actores, que es un Caso de Uso donde ingresan más de dos actores y estos tienen un rol común.

Luego de evaluar los patrones, los casos de uso son los siguientes:

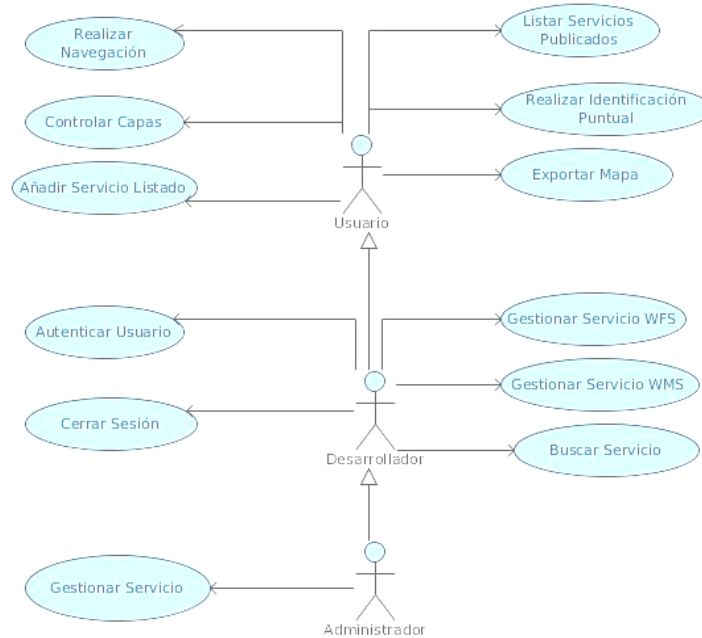


Figura 2: Diagrama de Casos de Uso del Sistema

5.1. Actores del sistema

Usuario: Persona que accede a las funcionalidades básicas del sistema, sin tener privilegios de escritura.

Desarrollador: Persona que se autentica para hacer uso de las funcionalidades y gestión de los servicios que proporciona una IDE.

Administrador: Persona que crea programas y aplicaciones que se implementan en el geoportal, es el responsable de asignar roles según los privilegios que se establezcan, fortaleciendo el nivel de

²⁸CRUD (Creating, Reading, Updating, Deleting por sus siglas en inglés)

seguridad del sistema, desde el que se puedan ver y utilizar los servicios y datos. Cumple además con la responsabilidad de revisar los servicios que son incorporados por los desarrolladores.

5.2. Descripción textual del Caso de Uso Gestionar Servicios de Mapas en la Web



Figura 3: Diagrama del Caso de Uso Gestionar Servicio WMS

Caso de Uso	Gestionar Servicio WMS	
Actores	Desarrollador	
Propósito	Este caso de uso permite que el desarrollador pueda adicionar, modificar, listar y eliminar Servicio WMS.	
Resumen	El caso de uso inicia cuando un desarrollador desea adicionar, modificar, listar o eliminar Servicio WMS en la aplicación, termina cuando el desarrollador realiza la(s) acción(es) y el sistema se actualiza una vez modificado.	
Precondiciones	El usuario debe estar autenticado. (RF 1)	
Referencias	RF 14, RF 15, RF 16, RF17	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el desarrollador selecciona en el geoportal la opción "Servicios de Mapa" (Ver Interfaz1).	2. El sistema muestra un formulario para que el desarrollador realiza la operación deseada (Ver Interfaz 1).	
3. El usuario selecciona la opción deseada: <ul style="list-style-type: none"> ◆ Adicionar Servicio de Mapas en la Web ◆ Modificar Servicio de Mapas en la Web 	4. El sistema muestra una interfaz según la opción seleccionada por el desarrollador. <ul style="list-style-type: none"> ◆ Si selecciona "Adicionar Servicio de Mapas en la Web", ver sección 	

<ul style="list-style-type: none"> ◆ Listar Servicio de Mapas en la Web ◆ Eliminar Servicio de Mapas en la Web 	<p>“Adicionar Servicio de Mapas en la Web”.</p> <ul style="list-style-type: none"> ◆ Si selecciona “Modificar Servicio de Mapas en la Web”, ver sección “Modificar Servicio de Mapas en la Web”. ◆ Si selecciona “Listar Servicio de Mapas en la Web”, ver sección “Listar Servicio de Mapas en la Web”. ◆ Si selecciona “Eliminar Servicio de Mapas en la Web”, ver sección “Eliminar Servicio de Mapas en la Web”.
	<p>5. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el desarrollador y actualiza la aplicación.</p>



Interfaz 1

Sección “Adicionar Servicio de Mapas en la Web”

Acción del Actor	Respuesta del Sistema
<p>1. El desarrollador selecciona la opción Adicionar Servicio de Mapas en la Web (Ver</p>	<p>2. El sistema muestra una ventana con los campos a llenar para adicionar el</p>

<p>Interfaz 2).</p>	<p>servicio (Ver Interfaz 3):</p> <ul style="list-style-type: none"> ◆ Título (Formato: Alfanumérico, Obligatorio: Si) ◆ Descripción (Formato: Alfanumérico, Obligatorio: No) ◆ Url (Formato: Alfanumérico, Obligatorio: Si) ◆ Temática (Formato: Alfanumérico, Obligatorio: Si) ◆ Localización (Formato: Alfanumérico, Obligatorio: Si)
<p>3. El desarrollador llena los campos.</p>	<p>4. El sistema verifica los datos, en caso de no tener errores, actualiza y adiciona el servicio.</p>

Servicios de Mapas

Lista **Agregar servicio** Temáticas Localizaciones de servicios

Seleccione el tipo de servicio para crear uno nuevo

WFS
Web Feature Service

WMS
Web Map Service

Interfaz 2

Lista **Agregar servicio** Temáticas Localizaciones de servicios

Parámetros básicos

Los siguientes parámetros de configuración son aplicables a todos los tipos de servicios

Título del servicio *

El título del servicio, que aparecerá en cada interfaz donde sea expuesto

Descripción del servicio

La descripción del servicio, que aparecerá en el área de administración

URL del servicio *

Ámbitos de servicios

Los siguientes parámetros de configuración son aplicables a todos los tipos de servicios

Temática del servicio *

Residencia

La temática del servicio

Localización del servicio *

Facultad Regional Artemisa

La localización del servicio

Parámetros específicos del servicio

Los siguientes parámetros de configuración son específicos a cada tipo de servicio

Guardar Cancelar

Interfaz 3

Flujos Alternos

Acción del Actor	Respuesta del Sistema
3. El desarrollador introduce datos incorrectos.	4. El sistema lanza un error y muestra los campos en blanco para que sean nuevamente llenados.
3. El desarrollador deja campos obligatorios sin llenar.	4. El sistema lanza un error y muestra los campos en blanco para que sean

	nuevamente llenados.
Sección “Modificar Servicio de Mapas en la Web”	
Precondición	El desarrollador debe haber listado el servicio inicialmente.
Acción del Actor	Respuesta del Sistema
1. El desarrollador selecciona la opción Editar (Ver Interfaz 4).	2. El sistema muestra una ventana con los campos posibles a modificar (Ver Interfaz 5) <ul style="list-style-type: none"> ♦ Título (Formato: Alfanumérico, Obligatorio: Si) ♦ Descripción (Formato: Alfanumérico, Obligatorio: No) ♦ Url (Formato: Alfanumérico, Obligatorio: Si) ♦ Temática (Formato: Alfanumérico, Obligatorio: Si) ♦ Localización (Formato: Alfanumérico, Obligatorio: Si)
3. El desarrollador llena los campos.	4. El sistema verifica los datos, en caso de no tener errores, actualiza y modifica el servicio.



Interfaz 4

Edit service

Parámetros básicos

Los siguientes parámetros de configuración son aplicables a todos los tipos de servicios

Título del servicio *

Nombre único o de máquina para la localización: *sdsdsdsds*
El título del servicio, que aparecerá en cada interfaz donde sea expuesto

Descripción del servicio

La descripción del servicio, que aparecerá en el área de administración

URL del servicio *

La URL del servicio

Ámbitos de servicios

Los siguientes parámetros de configuración son aplicables a todos los tipos de servicios

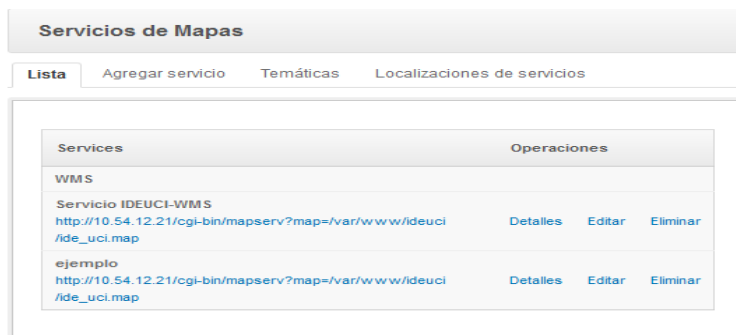
Interfaz 5

Flujos Alternos

Acción del Actor	Respuesta del Sistema
3. El desarrollador introduce datos incorrectos.	4. El sistema lanza un error y muestra los campos en blanco para que sean nuevamente llenados.
3. El desarrollador deja campos obligatorios sin llenar.	4. El sistema lanza un error y muestra los campos en blanco para que sean nuevamente llenados.

Sección “Listar Servicio de Mapas en la Web”

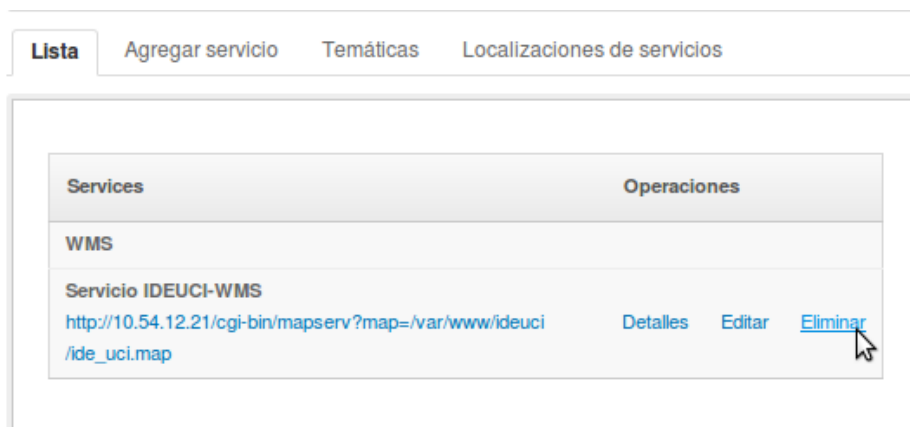
Acción del Actor	Respuesta del Sistema
1. El desarrollador selecciona la opción Listar Servicio de Mapas en la Web (Ver interfaz 6).	2. El sistema muestra una ventana con los servicios y las operaciones que existen.



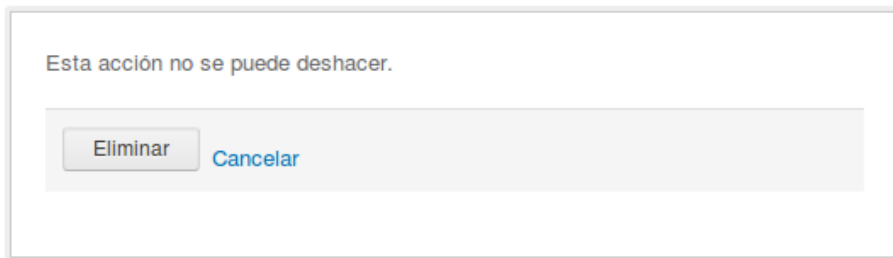
Interfaz 6

Sección “Eliminar Servicio de Mapas en la Web”

Precondición	El usuario debe haber listado el servicio inicialmente.	
Acción del Actor	Respuesta del Sistema	
1. El desarrollador selecciona la opción Eliminar (Ver Interfaz 7).	2. El sistema muestra una ventana verificando que el desarrollador desea eliminar el servicio (Ver Interfaz 8).	
3. El desarrollador confirma la eliminación del servicio.	4. El sistema elimina el servicio de la base de datos.	



Interfaz 7



Interfaz 8

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El desarrollador cancela la eliminación del servicio.	4. El sistema no elimina el servicio y muestra el estado inicial del servicio.
Poscondiciones	El sistema muestra la ventana actualizada según una las acciones del desarrollador.

Tabla 1: Descripción del Caso de Uso Gestionar Servicio WMS

6. Conclusiones parciales del capítulo

A partir de las necesidades identificadas en la UCI, se definieron las funcionalidades que debe aportar el sistema. Se tienen en cuenta las características no funcionales a las que debe responder la aplicación como: la apariencia e interfaz externa, la usabilidad, disponibilidad, eficiencia y seguridad. La utilización de patrones de casos de usos, permitió englobar los requisitos funcionales en casos de usos para un mejor trabajo con ellos, de manera que facilitó realizar la descripción detallada de las funcionalidades, ofreciéndoles a los usuarios los pasos a seguir para interactuar con la misma.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN

1. Introducción

En el presente capítulo se realiza el diseño para facilitar la implementación de los requisitos funcionales definidos en el capítulo anterior. Se analiza para la selección de los patrones arquitectónicos y de diseño los que emplea el CMS Drupal, debido que la aplicación forma parte de una personalización del mismo. Se generan los artefactos ingenieriles de acuerdo al diseño e implementación, entre estos se encuentran: la vista lógica, los diagramas de clases del diseño, diagrama entidad- relación, diagramas de componentes y modelo de despliegue.

2. Estilos arquitectónicos

2.1. Estilo Llamada y Retorno.

El Estilo Llamada y Retorno permite que el diseñador del software obtenga una estructura del programa que resulte fácil de manipular, modificar y cambiar de tamaño. Enfatiza la modificabilidad y la escalabilidad. Es un estilo generalizado para las arquitecturas del programa principal y subrutinas, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objetos y los sistemas jerárquicos en capas.

Drupal como CMS configurable refleja la estructura del lenguaje de programación. Se basa en la abstracción de procedimientos, funciones y métodos. De manera que este estilo arquitectónico brinda flexibilidad en la comunicación entre las funcionalidades de la aplicación web y establece una jerarquía que garantiza una organización y coordinación entre sus componentes.

3. Arquitectura de Software

La Arquitectura de Software es la organización fundamental de un sistema. Se representa a través de sus componentes, relaciones y restricciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Se concentra en requisitos no funcionales que son satisfechos mediante los modelos y diseños de la aplicación. Un diseño correcto de la Arquitectura del sistema es esencial para el éxito del proyecto (25).

El geoportal IDEUCI se realiza sobre el gestor de contenido Drupal v 7.9, de manera que para la selección de una arquitectura de software escalable se analiza su arquitectura.

La estructura de Drupal está sustentada en la base por el Sistema Operativo. Sobre este se asientan los tres componentes o tecnología fundamentales que utiliza Drupal, Apache como servidor web, el intérprete de PHP y en este caso, PostgreSQL como sistema Gestor de Base de Datos (38).

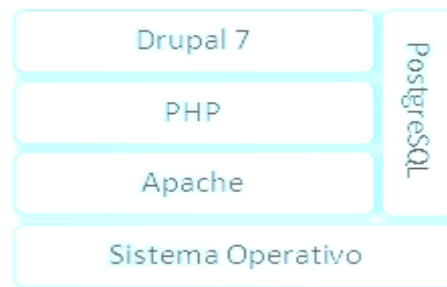


Figura 4: Estructura de Drupal

Dentro de los principales componentes de la arquitectura de Drupal están: núcleo, entidades, tipos de contenido, módulos, bloques, menús, temas, permisos de usuarios. Este gestor de contenido utiliza para su funcionamiento el patrón arquitectónico N-Capas y en cada capa hace uso del patrón Modelo – Vista- Controlador. A continuación se detallan estas arquitecturas.

3.1. Patron Arquitectónico N – Capas

El patrón N-capas o arquitectura en capas, descompone una aplicación en N cantidad de sub-tareas donde cada una de estas constituye un nivel de abstracción y se encuentran completamente distribuidas. En una arquitectura con capas, el usuario interactúa con la que se encuentra más arriba y esta interactúa con la que tiene debajo y así sucesivamente hasta llegar al fondo. A continuación se muestra una imagen con las capas que componen al CMS Drupal.

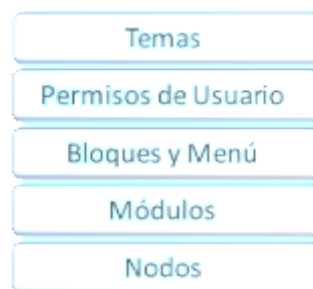


Figura 5: Arquitectura de Drupal

La capa base está formada por los **nodos** que son los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio web crece, lo va haciendo el número de nodos.

La segunda capa lo representan los **módulos**, estos son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal, permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web.

La tercera capa son los **bloques y menú**, estos permiten estructurar y organizar los contenidos en la página web. Son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

La capa **permisos de usuarios** dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario. Por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada.

La última capa la componen los **temas**, esta establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se desea es renovar la apariencia de un sitio web.

3.2. Patrón Arquitectónico Modelo- Vista- Controlador (MVC)

El Modelo Vista Controlador es un patrón arquitectónico que separa los datos de una aplicación, la interfaz de usuario y la lógica del negocio en tres componentes distintos. El sistema en desarrollo pertenece a la capa módulo dentro de la arquitectura del CMS, este utiliza el patrón MVC debido a que cada componente realiza una función específica que permite la interacción entre estos y en caso de modificación o ausencia de alguno el sistema no se afecte completamente.



Figura 6: Esquema del MVC

El modelo es el responsable de acceder a la capa de almacenamiento de datos, define las reglas de negocio, lleva un registro de las vistas y controladores del sistema y si se está en presencia de un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

La vista presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. Son responsables de recibir datos del modelo y los muestra al usuario, tienen un registro

de su controlador asociado y pueden brindar el servicio de actualización, para que sea invocado por el controlador o por el modelo.

El controlador responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo o a las vistas. Es el responsable del correcto funcionamiento de los eventos a responder por las peticiones de los usuarios.

Drupal está estructurado por temas de interfaces de interacción para con los usuarios, los cuales se pueden descargar de Internet o simplemente crearlos como plantillas en PHP, HTML y CSS. Posee una capa de abstracción de base de datos implementada y soportada para PostgreSQL, aunque puede añadir soporte para varias bases de datos. Los módulos que contienen algoritmos en PHP que proporcionan el funcionamiento del sistema, actúan como páginas servidoras y separan la interfaz gráfica de la información.

En la capa módulos se cuenta con tres componentes principales: *.install*, *.info* y *.module*. El archivo *.install* permite la conexión con la capa de acceso a datos (modelo), el *.info* brinda la información necesaria referente al módulo, dígame nombre, descripción, versión y dependencias (vista). El *.module* representa el componente controlador debido que es donde se implementan las funcionalidades que debe brindar la aplicación. Mediante la interrelación de estos componentes por cada capa de la arquitectura, se establece el funcionamiento de Drupal.

El uso del patrón Modelo Vista Controlador permite tener diferentes vistas para un mismo modelo, además de construir nuevas vistas sin necesidad de modificar el modelo subyacente y proporciona un mecanismo de configuración a componentes muchos más tratables para la implementación.

4. Patrones de Diseño

4.1. Patrones GRASP

Los patrones GRASP²⁹ son patrones generales de software para asignación de responsabilidades.

Indica el nivel de responsabilidad de la creación de un objeto o la implementación de un método, de manera que se conceda toda la información necesaria para crear la aplicación. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada.

Para identificar el responsable de la creación de instancias, módulos u objetos, se utiliza el patrón **creador** en los componentes *.modules* que permiten tener la información necesaria que se almacena o maneja por los módulos que se tienen o los que se crean. Esto facilita el

²⁹GRASP (General Responsibility Assignment Software Patterns por sus siglas en inglés)

mantenimiento, reutilización, mayor claridad y visibilidad en diseño de la aplicación.

Se utiliza el patrón **controlador** en la implementación en los componentes *.modules* porque actúan como intermediario entre una determinada interfaz y el algoritmo que implementan, de tal forma que recibe los datos del usuario, procesa la información y realiza llamadas a diferentes componentes para dar la respuesta. De manera que aumenta la reutilización de código y a la vez se tiene mayor organización y control.

Para que la información almacenada en un módulo sea lo menos dependiente entre sí y coherente se utilizan los patrón **alta cohesión y bajo acoplamiento**, de tal forma que los archivos estén relacionados y en caso de producirse una modificación en alguno, se tenga la mínima repercusión posible en el resto de estos, potenciando la reutilización, y disminuyendo la dependencia entre estos.

Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento. De esta forma se tiene menor dependencia y se especifican los propósitos de cada objeto en el sistema.

4.2. Patrones GoF

Los patrones GoF³⁰ o “Pandilla de los Cuatro” representan una descripción de las clases y los objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular. Este se clasifican en 3 categorías: los creacionales, estructurales y de comportamiento.

El sistema cuenta con un archivo controlador *.module* por cada módulo que según las peticiones de los usuarios delega responsabilidades para que continúe el flujo de actividades. De esta forma se demuestra el uso del patrón **Singleton (Instancia única)** que pertenece a los patrones **creacionales**. Se garantiza que sólo se tenga una instancia que proporcione un punto de acceso global y permita asociar diferentes módulos con un conjunto de funciones distintas. Esto facilita que la conexión sea única, accesible y reutilizable para todos.

El uso del patrón **Decorator (Envoltorio)** dentro de los **estructurales**, permite añadir funcionalidades a un objeto dinámicamente, posibilitando no tener que crear sucesivos módulos incorporando la nueva funcionalidad, sino implementar las funciones y asociarla al módulo principal. Se evidencia este patrón al añadir bloques y menús a una interfaz sin tener que modificar otras.

³⁰GoF(Gang of Four por sus siglas en inglés)

En cuanto a los patrones de **comportamiento** se utiliza el patrón **Observer (Observador)**, que define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Se evidencia cuando una modificación es hecha a un vocabulario en el sistema de taxonomía, el hook taxonomy es llamado en todos los módulos que lo implementan. Mediante la aplicación del hook, se han registrado como observadores del objeto vocabulario; cualquier cambio a esto puede entonces actuar como es apropiado.

El patrón **Chain of Responsibility (Cadena de responsabilidad)**, permite establecer la línea que debe llevar los mensajes para que los objetos realicen la tarea indicada. Se evidencia el uso de este patrón en la implementación del hook_menu () ya que en cada solicitud de la página, el menú del sistema determina si hay un módulo que la gestione, si el usuario tiene acceso a los recursos solicitados, y que la función sea llamada para realizar su objetivo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud y si el elemento del menú no puede manejar la petición, se pasa en cadena. Esto continúa hasta que un módulo se encarga de la petición o un módulo niega el acceso para el usuario, o la cadena se ha agotado.

El patrón **Command (Orden)** encapsula una operación en un objeto, permitiendo ejecutar la operación sin necesidad de conocer el contenido de la misma. Los hooks definen las operaciones que se pueden realizar en o por un módulo. La implementación de estos en los módulos del sistema se centra en realizar la tarea en particular que invoca el hook, sin necesidad de conocer el contenido del mismo. De manera que aumenta la reutilización de código y reduce el número de funciones a implementar.

5. Vistas del sistema

5.1. Vista Lógica

La vista lógica describe la estructura lógica del sistema. Abarca la descripción de la arquitectura y presenta los elementos estructurales y del comportamiento predominante.

5.1.1. Elementos del modelo arquitectónicamente significantes

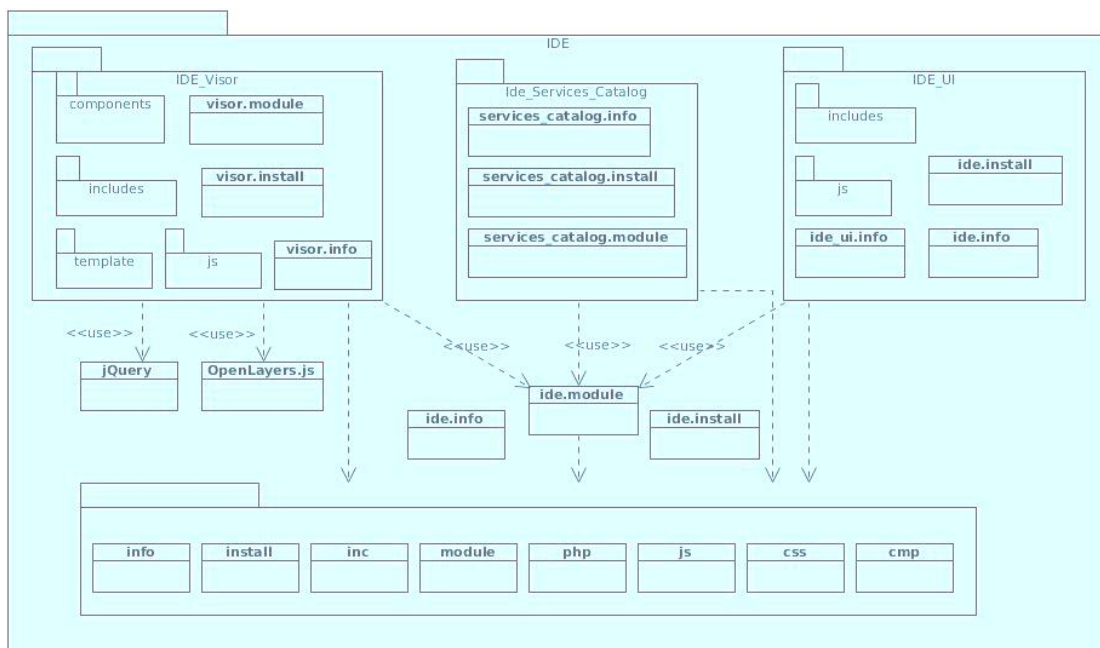


Figura 7: Diagrama Vista Lógica

La estructura lógica del sistema IDEUCI está compuesta por el paquete llamado “IDE” que contiene una personalización de los módulos brindados por el CMS Drupal y con los ficheros necesarios para la comunicación entre los módulos creados (info, install, module, js, php, css, inc, cmp). A su vez módulo IDE incluye los módulos “IDE_UI”, “IDE_Visor e “IDE_Service_Catalog”. IDE_UI posibilita la administración y configuración. IDE_Visor que permite la configuración de los componentes haciendo uso de la librería jQueryUI e implementación de funcionalidades que brindará el visor del sistema utilizando la librería OpenLayers.js para la visualización de los mapas. IDE_Service_Catalog brinda el servicio catálogo para realizar la búsqueda y/o consulta de los servicios que están publicados en la aplicación.

6. Visión general de la arquitectura

La aplicación cuenta con 3 capas lógicas. A continuación se hace una descripción de cada capa y la forma en que interactúan entre sí.

Interfaz: En esta capa están implementadas todas las interfaces gráficas con las que interactúa el usuario, y las interfaces de interacción con otros sistemas. Se desarrolló aplicando tecnologías como PHP, HTML y Java Script.

Negocio: En esta capa se integra la librería OpenLayers.js que permite la visualización de los mapas en el visor. El servidor de mapas MapServer haciendo uso del archivo .map lee y configura las capas del mapa UCI incorporando los servicios WMS y WFS. Se crea el módulo “ide” con las funcionalidades necesarias para lograr la interoperabilidad entre los sistemas que posibilitan el trabajo con el visor, administración de contenido, información geográfica y servicios.

Base de Datos: En esta capa se encuentra la base de datos de la aplicación y contiene en las tablas los datos que se almacenan según peticiones realizadas al usuario, además de la información de consulta. El servidor de bases de datos utilizado es PostgreSQL.

7. Modelo de diseño

Una vez que se analizan y especifican los que requisitos funcionales y no funcionales, el diseño es la primera tarea que se realiza para la construcción y verificación del software. Este modelo permite comprender el comportamiento de las principales funcionalidades del sistema.

7.1. Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las dependencias, métodos, clases del software y las interfaces en una aplicación. Brinda una visión general de la solución, lo que resulta más eficiente para efectuar cambios y a su vez detectarlos sin problemas.

A continuación se muestra el diagrama de clases del diseño del Caso de Uso Gestionar Servicio WMS.

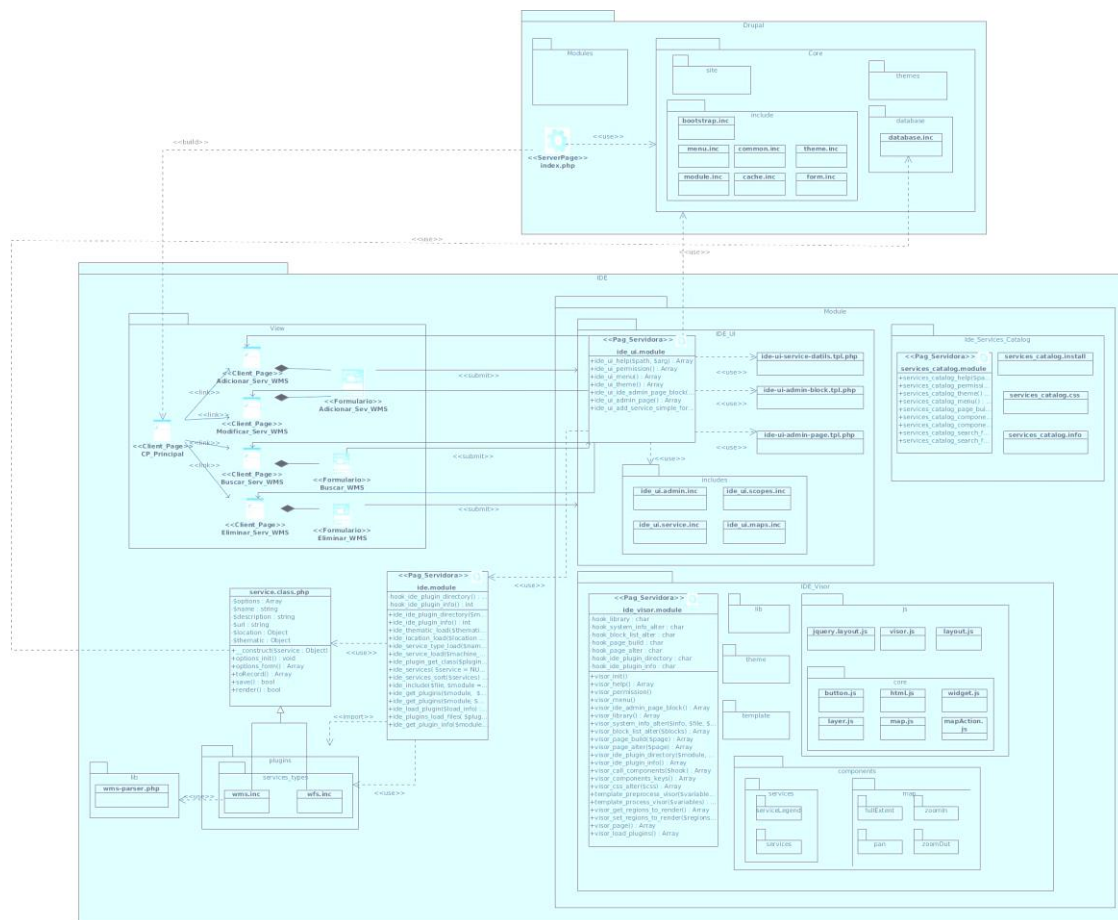


Figura 8: Diagrama de Clase del Diseño Gestionar Servicio WMS

7.1.1. Descripción del diseño

Para la comprensión de la estructura y funcionamiento del diagrama, a continuación se explica el mismo.

El paquete de **Drupal** está contenido fundamentalmente por:

Modules: Paquete que almacena todos los módulos que permiten las distintas funcionalidades del CMS, cuando desee incorporar un nuevo módulo sólo se tiene que incorporar en dicha carpeta.

Index.php: Página que recibe todas las peticiones que realizan los usuarios, enviándolas posteriormente al bootstrap.inc para que continúe el flujo de actividades.

Core: El mismo está compuesto por los paquetes *includes*, *themes*, *database* y *sites* que permiten el correcto funcionamiento del framework. El paquete includes contiene un conjunto de ficheros imprescindibles para el funcionamiento del Drupal como es el caso del **bootstrap.inc**

el cual es el encargado de activar los módulos según las peticiones que estén solicitando los usuarios, el fichero **database.inc** que proporciona las funcionalidades de acceso a la base de datos. El paquete *themes* almacena todas las plantillas de los diseños que se le quieran aplicar a la aplicación web. El paquete *sites* permite la personalización de los módulos que contiene Drupal, además de la incorporación de otros por la comunidad.

Dentro del paquete **IDE** se pone en práctica el uso del Patrón Modelo- Vista- Controlador. Este paquete está compuesto por el paquete **View** que contiene todas las páginas clientes y los formularios que interactuarán con el usuario directamente y representa la vista del sistema. En el mismo agrupa el paquete **Module** que incluyen los paquetes de los módulos *ide_ui*, *ide_visor* e *ide_service_catalog* y representa el controlador del sistema. La clase **service.class.php** es la encargada de gestionar la información requerida de los servicios a través del componente *database.inc* para brindar los datos una vez realizada las peticiones.

Una vez realizada la(s) petición(es) por el cliente (Adicionar, Modificar, Listar y/o Eliminar Servicio WMS), el componente controlador **ide_ui.module** es el responsable de gestionar los servicios, obteniendo de **ide.module** el tipo de servicio, la localización y temática, utilizando la clase **ide_ui_service.datils.tpl.php** y los componentes *.inc* que se encuentran dentro del paquete *includes* que tienen implementadas funcionalidades para la creación de servicios (**ide_ui.services.inc**), obtención de las capas del mapa UCI (**ide_ui.maps.inc**), administración de las categorías de los servicios (**ide_ui.scopes.inc**) y administración de los permisos de usuarios (**ide_ui.admin.inc**). Una vez que se obtenga la respuesta, esta es enviada al controlador principal, actualizando o modificando los datos mediante la clase **service.class.php** para brindar la respuesta al usuario.

8. Modelo de datos

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que describen los datos. Constituye una herramienta que facilita la interpretación y representación en forma de datos y permite definir formalmente las estructuras permitidas y sus restricciones a fin de representar los datos de los elementos básicos en el diseño de una base de datos.

8.1. Diseño de la base de datos

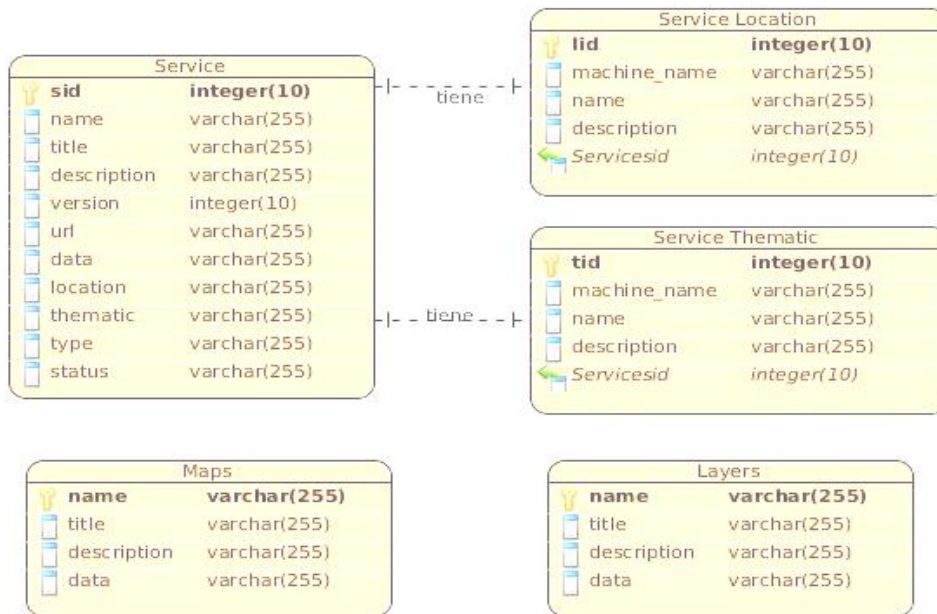


Figura 9: Diagrama de Entidad-Relación

8.2. Descripción de las tablas

Service: Almacena el nombre, título, versión, una breve descripción, url, datos, localización, temática, estado, tipo de servicio y un identificador que posibilita que solo exista una ocurrencia única en la base de datos.

Service_Location: Guarda información detallada sobre la ubicación del servicio que se desea añadir para tener bien registrado la procedencia de este.

Service_Thematic: Guarda los valores específicos sobre la temática del servicio, lo que posibilita tener palabras claves para realizar la búsqueda de este.

Maps: Almacena los mapas que se añaden al visor.

Layout: Almacena las capas de los mapas que se adicionan al visor.

9. Vista de despliegue

Esta vista suministra una base para la comprensión de la distribución física del sistema a través de nodos. Suele utilizarse cuando el sistema está distribuido y hay una traza directa del modelo de implementación, puesto que cada componente físico debe estar almacenado en un nodo. Esto incluye también la asignación de tareas provenientes de la vista de procesos en los nodos.

9.1. Diagrama de Despliegue

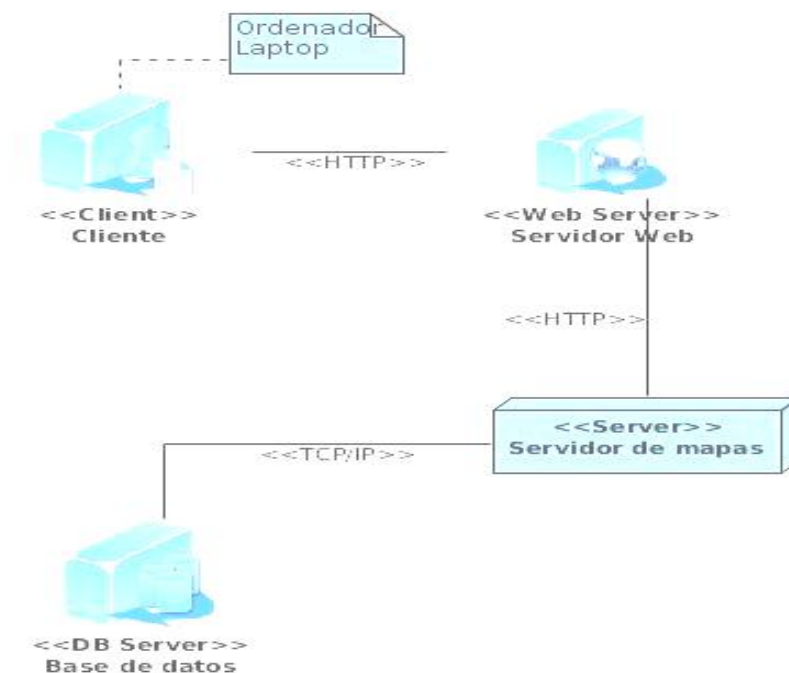


Figura 10: Diagrama de despliegue

9.1.1. Descripción del diagrama de despliegue

Se cuenta con un nodo cliente compuesto por una PC o laptop con un procesador Pentium 3, 256 MB de memoria RAM como mínimo, debido a que se requiere al menos 100 MB de disco duro. El nodo cliente se comunica mediante el protocolo HTTP al servidor web (Web Server) que realizará la construcción de interfaces de usuarios, el procesamiento de datos, y el control de flujo. Este a su vez se comunica haciendo uso del protocolo HTTP al Servidor de Mapas (Server) que contiene los ficheros de configuración y accederá por el protocolo TCP/IP al servidor de base de datos (DB Server) que almacena los datos de la aplicación. La base de datos puede estar distribuida en otros nodos y conectadas entre ellas.

10. Diagrama de componentes

El diagrama de componentes representa cómo un sistema de software es dividido en componentes, muestra las dependencias y las relaciones entre estos. Muestra la organización y describe los elementos físicos del sistema que incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes utilizados para modelar la vista estática y dinámica de la aplicación.

Los componentes *.module* de cada módulo contienen varias funcionalidades que responden a más de un requisito funcional de la aplicación, estos a su vez hacen uso de otros componentes integrados al

módulo al que pertenecen para facilitar la implementación. Como se utiliza un componente controlador por cada módulo que realiza más de un caso de uso, se diseña el diagrama de componentes por módulo. A continuación se ofrece el diagrama de componentes del sistema.

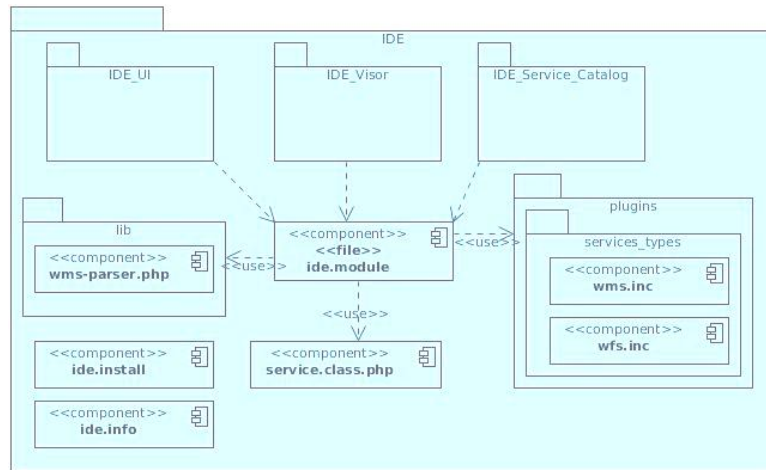


Figura 11: Diagrama de componentes del módulo ide

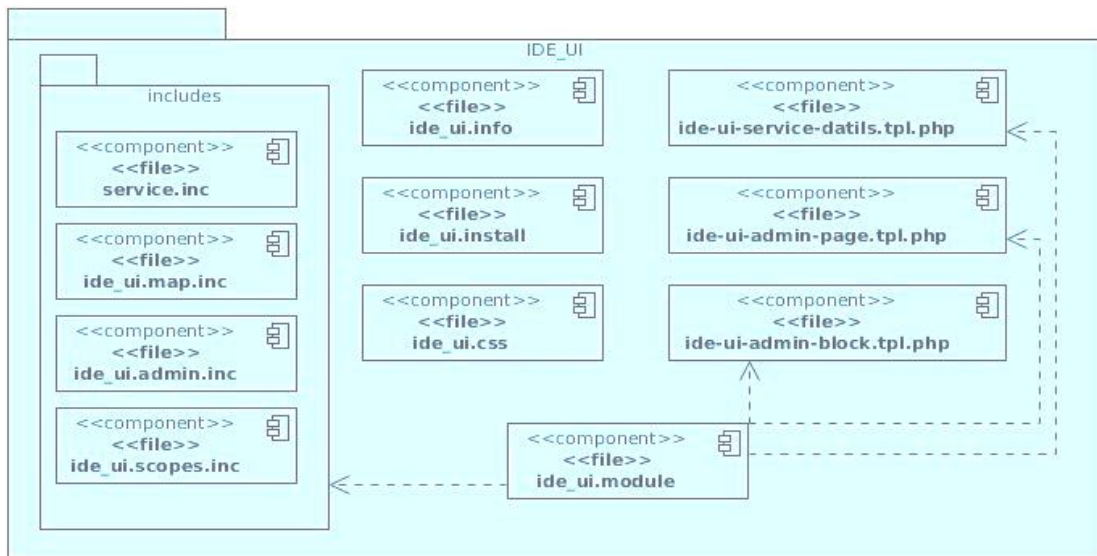


Figura 12: Diagrama de componente del módulo ide_ui

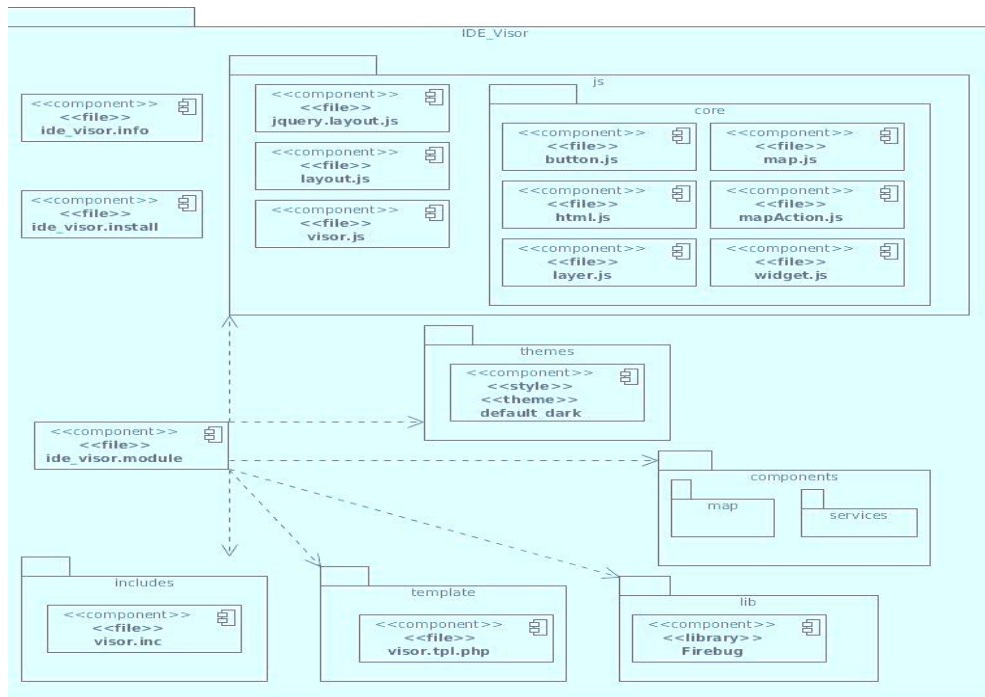


Figura 13: Diagrama de componente del módulo ide_visor

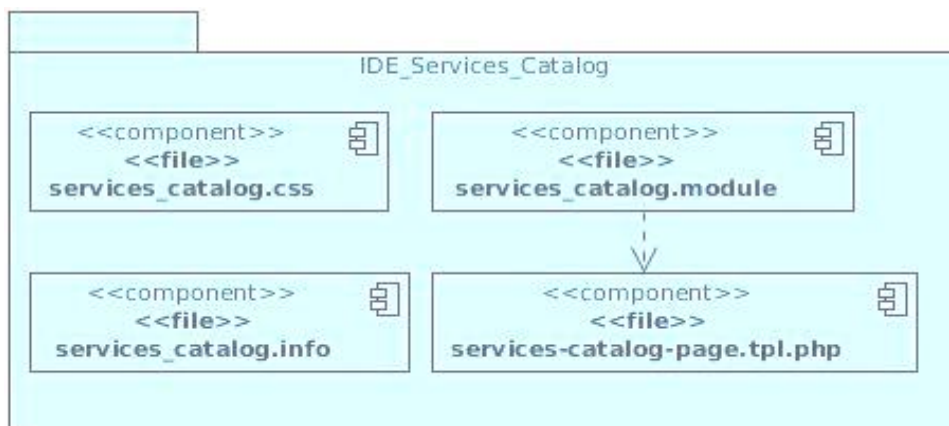


Figura 14: Diagrama de componente del módulo ide_servie_catalog

11. Estilos de Codificación

Los estilos de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

Dentro de las normas de programación de Drupal se establece que los archivos principales (*.install*, *.info* y *.module*) tengan un mismo nombre para poder ser interpretado y analizado por el CMS. Drupal

se basa en implementación de hook que son el mecanismo que provee para interactuar con los distintos procesos que se ejecutan en un sitio web. Dentro de los hook más empleados en la aplicación se encuentran:

help: Brinda una ayuda de la páginas y módulos creados.

theme: Posee diferentes temas que definen la forma en la que se muestran los contenidos de un módulo.

permission: Suministra los permisos que el módulo define, estos son basados en los roles establecidos a los usuarios.

block: Permite incorporar uno o varios bloques para un módulo.

menu: Define los elementos de menú y respuestas de las peticiones realizadas a la página.

form: Muestra un formulario de edición de nodos. Permite la administración y configuración de las secciones añadidas por otros módulos.

init: Es ejecutado en el comienzo de la solicitud de página y permite realizar tareas de configuración para las solicitudes de página no almacenados en caché.

11.1. Principios Generales

La codificación del sistema está regida por los siguientes principios generales:

- ◆ Los nombres de cada uno de los elementos del programa deben ser significativos, es decir, debe describir el uso del elemento.
- ◆ No manejar en los programas más de una instrucción por línea.
- ◆ Declarar las variables en líneas separadas.
- ◆ Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- ◆ La mayoría de los elementos se deben nombrar usando sustantivos.
- ◆ Para las llaves que definen el cuerpo de una función, sangre un nivel.
- ◆ Para distinguir palabras dentro del nombre debe emplearse una letra mayúscula o un guión bajo (_), sin mezclar ambas formas en un mismo programa.
- ◆ Se realizan comentarios cuando se declaren variables, métodos, llamadas a función o inclusión de archivos que no sean obvios dentro de los módulos creados, este incluye el objetivo, o problema que resuelve, el algoritmo.

- ◆ Cada función debe tener un encabezado que contenga objetivo de la función y no descripción del procedimiento.
- ◆ Explicación de uso de argumentos (parámetros) no obvios y de los de uso de valores devueltos (de retorno).
- ◆ Se realiza un salto de línea cuando termina la sentencia, es decir, luego del punto y coma.
- ◆ Se utiliza espacios en blanco para mejorar la legibilidad del código.
- ◆ Evitar las líneas con demasiados caracteres, de manera que cuando el código no se descifre bien se debe reorganizar usando los principios descritos anteriormente.
- ◆ Cuando se le asigna un texto literal (sin contenido de variables) se utilizan comillas simples.
- ◆ Para concatenar strings se utiliza el operador "." (punto), con un espacio entre medio para mejorar la lectura.

A continuación se muestra un ejemplo de código de la clase `ide_ui.module` para evidenciar los principios establecidos:

```

/**
 * Implementation of hook_theme()
 */
function ide_ui_theme()
{
  return array
  (
    'ide_ui_admin_page' => array
    (
      'template' => 'ide-ui-admin-page',
      'variables' => array( 'blocks' => NULL, )
    ),
    'ide_ui_admin_page_block' => array
    (
      'render element' => 'elements',
      'template' => 'ide-ui-admin-block',
      'variables' => array( 'block' => array(), )
    ),
    'ide_ui_servide_details' => array
    (
      'variables' => array( 'fields' => array(), 'xml' => NULL, ),
      'template' => 'ide-ui-service-datils',
    ),
  );
}

```

Figura 15: Ejemplo de código

12. Conclusiones parciales del capítulo

En el presente capítulo los artefactos generados representan la estructura e interrelación de los componentes para el correcto funcionamiento del sistema. Mediante la representación de las vistas lógicas, de arquitectura, de implementación y despliegue se obtiene una visión general sobre los principales objetos, nodos y ficheros que se presentan, propiciando una mejor comprensión de las relaciones y la comunicación entre estos. El uso de la arquitectura N-Capas posibilita la que fluya información entre las capa que posee el CMS Drupal y a su vez cada capa hace uso de la arquitectura Modelo- Vista -Controlador para la implementación del sistema.

CAPÍTULO 4: VALIDACIÓN DEL SISTEMA

1. Introducción

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requisitos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante las pruebas. El proceso de pruebas, sus objetivos, métodos y técnicas usados se describen en el plan de prueba. En el presente capítulo se realizan los casos de prueba para validar el funcionamiento de la aplicación.

2. Pruebas del sistema

Las pruebas de software son un conjunto de actividades dentro del desarrollo de software en el proceso de aseguramiento de la calidad, no aseguran la ausencia de errores; sólo puede demostrar que existen defectos en el software. Diseñar pruebas posibilita encontrar varios de errores con la mínima cantidad de esfuerzo y de tiempo.

Luego de realizar la implementación del sistema es necesario comprobar que este cumple con lo establecido inicialmente. De manera que hay que verificar el correcto funcionamiento de los requisitos funcionales establecidos. La estructura se valora para asegurarse de que entrega adecuadamente el contenido y función, que es extensible y puede sostenerse conforme se añade nuevo contenido o funcionalidad.

Para ello se realizan pruebas de contenido, que permiten evaluar tanto el ámbito sintáctico (ortografía, puntuación y gramática) como semántico (exactitud de la información, la consistencia de los objetos relacionados y la falta de ambigüedad). Se realizan pruebas de facilidad de uso, interfaz y navegabilidad para garantizar que cada interfaz de usuario está en correspondencia con las acciones seleccionadas, permite comprobar que la navegación se realiza sin que existan vínculos rotos, inadecuados o erróneos. Se realizan las pruebas de componentes que validan cada componente por separado, de manera que se prueba la independencia del modelo, de la vista y el controlador, para verificar su correcto funcionamiento de forma individual (37).

La configuración del API de Drupal por defecto está diseñada para ser seguro cuando se usa en los modos predeterminados. Temas como la inyección SQL, Cross Site Scripting, gestión de sesiones, Cross Site Request Forgeries, entre otros, tienen las soluciones estándares con el uso de funcionalidades establecidas.

Uno de los beneficios que aporta el uso de Drupal 7 es la seguridad debido a que establece permisos según los roles asignado a los usuarios. Para realizar esta prueba se definieron los roles que intervienen en el sistema: usuarios, desarrollador y administrador. A cada rol se le otorgó los permisos en correspondencia. Luego de haber hecho la prueba se obtuvo resultados satisfactorios pues cada usuario es capaz de realizar las funciones específicas de su rol y ninguna otra.

IDEUCI como módulo predefinido dentro de este CMS Drupal versión 7.9 hace uso de los algoritmos que garantizan la seguridad del sistema. Para evitar inyecciones de código SQL se realizan consultas dinámicas que almacenan la información en variables, se demuestra con el uso de las funciones `db_query ()`, `db_select ()`, `fields ()`, `condition ()`, `execute ()` y `range ()`.

Se realizan pruebas funcionales a la aplicación con el objetivo de demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. De ahí que se realiza las pruebas de caja negra, específicamente el método partición equivalente. Este método divide el dominio de entrada de un programa en clases de datos, a partir de las cuales deriva los casos de prueba. Cada una de estas clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada. Se diseñan los casos de pruebas que validan la entrada de los datos y la respuesta correcta del sistema.

3. Diseño de caso de Prueba del CU Gestionar Servicio WMS

3.1. Descripción General.

El caso de uso inicia cuando un usuario desea adicionar, modificar, listar o eliminar Servicios WMS para la aplicación y termina cuando el usuario realiza la(s) acción(es) y el sistema se actualiza una vez modificado.

3.2. Condiciones de Ejecución.

El usuario se encuentra autenticado en el sistema.

3.3. Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Adicionar Servicio WMS.	EC 1.1: Seleccionar la opción "Adicionar Servicio WMS".	El sistema muestra una ventana con los campos "Título", "Descripción", "Url", "Temática" y "Localización".
	EC 1.2: Introducción de los	El sistema valida los datos y añade el

	datos correctamente.	servicio WMS.
	EC 1.3: Introducción de los datos incorrectamente.	El sistema lanza un error y muestra nuevamente el formulario descrito en el escenario 1.1 con los campos vacíos para que el usuario vuelva a introducir los datos.
	EC 1.4: Se dejan campos obligatorios vacíos.	El sistema lanza un error muestra nuevamente el formulario descrito en el escenario 1.1 con los campos vacíos para que el usuario vuelva a introducir los datos.
SC 2: Modificar Servicio WMS.	EC 2.1: Seleccionar la opción "Modificar Servicio WMS".	El sistema muestra una ventana con los campos "Título", "Descripción", "Url", "Temática" y "Localización".
	EC 2.2: Introducción de los datos correctamente.	El sistema valida los datos y modifica el servicio WMS.
	EC 2.3: Introducción de los datos incorrectamente.	El sistema lanza un error y muestra nuevamente el formulario descrito en el escenario 2.1 con los campos vacíos para que el usuario vuelva a introducir los datos.
	EC 2.4: Se dejan campos obligatorios vacíos.	El sistema lanza un error y muestra nuevamente el formulario descrito en el escenario 2.1 con los campos vacíos para que el usuario vuelva a introducir los datos.
SC 3: Listar Servicio WMS.	EC 3.1: Seleccionar la opción "Listar Servicio	El sistema muestra una ventana con los todos los servicios WMS

	WMS".	existentes.
SC 4: Eliminar Servicio WMS.	EC 4.1: Seleccionar la opción "Eliminar Servicio WMS".	El sistema muestra una ventana de confirmación.
	EC 4.2: Confirma la eliminación del servicio.	El sistema elimina el servicio.
	EC 4.3: Cancela la eliminación del servicio.	El sistema no elimina el servicio y muestra el estado inicial del servicio.

Tabla 2: Escenario del Caso de Uso Gestionar Servicio WMS

3.4. Descripción de variable

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Título	Campo de texto	No	Solo contiene caracteres en mayúscula desde la A hasta la Z y en minúscula desde la a hasta la z.
2	Url	Campo de texto	No	Contiene caracteres en mayúscula desde la A hasta la Z y en minúscula desde la a hasta la z y los símbolo "?", "=", "&" para conformar la url.
3	Descripción	Campo de texto	No	Solo contiene caracteres en mayúscula desde la A hasta la Z y en minúscula desde la a hasta la z.
4	Temática	Campo de texto	No	Solo contiene caracteres en mayúscula desde la A hasta la Z y en minúscula desde la a hasta la z.
5	Localización	Campo de texto	No	Solo contiene caracteres en mayúscula desde la A hasta la Z y en minúscula desde la a hasta la z.

Tabla 3: Variables del Caso de Uso Gestionar Servicio WMS

3.5. Matriz de Datos

3.5.1. SC 1: Adicionar Servicio WMS

Escenario	Var. 1 Título	Var. 2 Url	Var. 3 Descripción	Var. 4 Temática	Var. 5 Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 1.1: Seleccionar la opción "Adicionar Servicio WMS".	NA	NA	NA	NA	NA	El sistema muestra una ventana con los campos "Título", "Descripción", "Url", "Temática" y "Localización".	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar en geoportal "Servicio de mapas". 2. Seleccionar la opción "Adicionar Servicio WMS".
EC 1.2: Introducción de los datos correctamente	V	V	V	V	V	El sistema valida los datos y añade el servicio WMS.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar en geoportal "Servicio de mapas". 2. Seleccionar la opción "Adicionar Servicio WMS".
EC 1.3: Introducción de los datos incorrectamente.	I	V	V	V	V	El sistema lanza un error y muestra nuevamente el formulario descrito en el escenario 1.1 con los campos vacíos para que el usuario vuelva a introducir los datos.	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar en geoportal "Servicio de mapas". 2. Seleccionar la opción "Adicionar Servicio WMS". 3. Seleccionar la opción "Aceptar".
	V	I	V	V	V			
	V	V	I	V	V			
	V	V	V	I	V			
	V	V	V	V	I			
	I	I	I	I	I			
EC 1.4: Se dejan campos	I	V	V	V	V	El sistema lanza un error	Satisfactorio	<ol style="list-style-type: none"> 1. Seleccionar en

obligatorios vacíos.	V	I	V	V	V	muestra nuevamente el formulario descrito en el escenario 1.1 con los campos vacíos para que el usuario vuelva a introducir los datos.		geoportal “Servicio de mapas”. 2. Seleccionar la opción “Adicionar Servicio WMS”. 3. Seleccionar la opción “Aceptar”.
	V	V	I	V	V			
	V	V	V	I	V			
	V	V	V	V	I			
	I	I	I	I	I			

Tabla 4: Matriz de datos de Adicionar Servicio WMS

3.5.2. SC 2 Modificar Servicio WMS

Escenario	Var. 1 Título	Var. 2 Url	Var. 3 Descripción	Var. 4 Temática	Var. 5 Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 2.1: Seleccionar la opción “Editar Servicio WMS”.	NA	NA	NA	NA	NA	El sistema muestra una ventana con los campos “Título”, “Descripción”, “Url”, “Temática” y “Localización”.	Satisfactorio	1. Seleccionar en geoportal “Servicio de mapas”. 2. Seleccionar la opción “Editar Servicio WMS”.
EC 2.2: Introducción de los datos correctamente.	V	V	V	V	V	El sistema valida los datos y modifica el servicio WMS.	Satisfactorio	1. Seleccionar en geoportal “Servicio de mapas”. 2. Seleccionar la opción “Editar Servicio WMS”. 3. Seleccionar la opción “Aceptar”.
EC 2.3: Introducción de los datos incorrectamente.	I	V	V	V	V	El sistema lanza un error y muestra nuevamente el formulario descrito en el escenario 2.1	Satisfactorio	1. Seleccionar en geoportal “Servicio de mapas”. 2. Seleccionar la opción “Editar
	V	I	V	V	V			
	V	V	I	V	V			

	V	V	V	I	V	con los campos vacíos para que el usuario vuelva a introducir los datos.		Servicio WMS”. 3. Seleccionar la opción “Aceptar”.
	V	V	V	V	I			
	I	I	I	I	I			
EC 2.4: Se dejan campos obligatorios vacíos.	I	V	V	V	V	El sistema lanza un error muestra nuevamente el formulario descrito en el escenario 2.1 con los campos vacíos para que el usuario vuelva a introducir los datos.	Satisfactorio	1. Seleccionar en geoportal “Servicio de mapas”. 2. Seleccionar la opción “Editar Servicio WMS”. 3. Seleccionar la opción “Aceptar”.
	V	I	V	V	V			
	V	V	I	V	V			
	V	V	V	I	V			
	V	V	V	V	I			
	I	I	I	I	I			

Tabla 5: Matriz de datos de Modificar Servicio WMS

3.5.3. SC 3 Listar Servicio WMS

Escenario	Var. 1 Título	Var. 2 Url	Var. 3 Descripción	Var. 4 Temática	Var. 5 Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
EC 3.1: Seleccionar la opción “Listar Servicio WMS”.	NA	NA	NA	NA	NA	El sistema muestra una ventana con todos los servicios WMS existentes.	Satisfactorio	1. Seleccionar en geoportal “Servicio de mapas”. 2. Seleccionar la opción “Listar Servicio WMS”.

Tabla 6: Matriz de datos de Listar Servicio WMS

3.5.4. SC 4 Eliminar Servicio WMS

Escenario	Var. 1 Título	Var. 2 Url	Var. 3 Descripción	Var. 4 Temática	Var. 5 Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
-----------	---------------	------------	--------------------	-----------------	---------------------	-----------------------	------------------------	---------------

EC 4.1: Seleccionar la opción "Eliminar Servicio WMS".	NA	NA	NA	NA	NA	El sistema muestra una ventana de confirmación.	Satisfactorio	1 Seleccionar en geoportal "Servicio de mapas". 2. Seleccionar la opción "Eliminar Servicio WMS".
EC 4.2: Confirma la eliminación del servicio.	V	V	V	V	V	El sistema elimina el servicio.	Satisfactorio	1. Seleccionar en geoportal "Servicio de mapas". 2. Seleccionar la opción "Eliminar Servicio WMS". 3. Seleccionar la opción "Aceptar".
EC 4.3: Cancela la eliminación del servicio.	NA	NA	NA	NA	NA	El sistema no elimina el servicio y muestra el estado inicial del servicio.	Satisfactorio	1. Seleccionar en geoportal "Servicio de mapas". 2. Seleccionar la opción "Eliminar Servicio WMS". 3. Seleccionar la opción "Cancelar".

Tabla 7: Matriz de datos de Eliminar Servicio WMS

3.6. Resultados de las Pruebas

Al realizar la técnica de partición equivalente se obtiene en una primera iteración inconformidades en las funcionalidades realizar identificación puntual, control de capas, añadir servicios WMS y WFS. Entre las inconformidades se encuentran errores en la identificación puntual de los objetos en el visor. En el control de capa muestra todo el mapa en un solo campo de selección en vez de las capas separadas para habilitarlas o deshabilitarlas. Al añadir el servicio WMS, para mostrarlo en el visor no reconoce las operaciones de la url. Por su parte al añadir el servicio WFS existen errores al exportar el fichero. Luego al corregir estos errores, se realizó una segunda iteración donde los resultados fueron satisfactorios. A continuación de muestra un gráfico haciendo referencia a los resultados obtenidos:

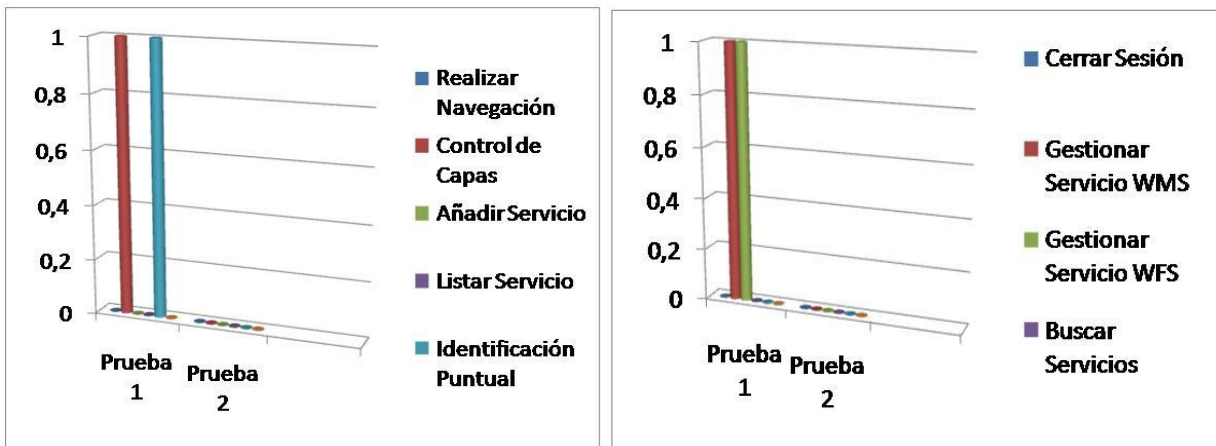


Figura 16: Resultados de las pruebas

4. Conclusiones Parciales

Al realizarse las pruebas funcionales al sistema, se verificó que la navegabilidad era correcta, que los componentes responden de acuerdo a lo establecido por separado y en conjunto, lo que representa la integridad e interrelación entre estos. Las pruebas de contenido e interfaz facilitaron que las interfaces de usuarios fuesen corregidas, se agregó la información que correspondía a cada menú según el vínculo realizado. Se verificó que la aplicación fuese accedida desde diversos puntos para comprobar las comunicaciones de redes. Para asegurar el correcto funcionamiento del sistema se realizaron los diseños de casos de pruebas, haciendo uso de la técnica partición equivalente perteneciente a las pruebas de caja negra. Se demostró mediante la ejemplificación del diseño de caso de prueba del caso de uso crítico Gestionar Servicio WMS, arrojando a resultados satisfactorios. En general la realización de pruebas permitió corregir los errores de la aplicación para obtener los resultados esperados.

CONCLUSIONES

A partir del aumento del uso de información geoespacial en la Universidad de las Ciencias Informáticas era necesario la realización de una aplicación que posibilitara la centralización, integración y provisión de esta información. Mediante la creación de la Infraestructura de Datos Espaciales para la UCI (IDEUCI) se cumplieron los objetivos planteados en un inicio, arribando a las siguientes conclusiones:

- ◆ IDEUCI posibilita centralizar, integrar y proveer la información geográfica de la universidad, facilitando el acceso, obtención y utilización de la misma por todos aquellos usuarios que la necesiten.
- ◆ Posibilita a la comunidad universitaria conocer los beneficios y potencialidades que ofrecen el uso de los servicios web en las aplicaciones que trabajen con la información geoespacial.
- ◆ El modelado del sistema desarrollado permite una correcta descripción técnica para la creación de futuras versiones.
- ◆ La selección de herramientas y tecnologías permite que el sistema fuese desarrollado libre de costo en términos de pago de licencia.
- ◆ IDEUCI representa un aporte a la comunidad de Drupal al incorporar un módulo que integra un visor que posibilita añadir de forma dinámica la(s) capa(s) del mapa(s) UCI, el uso, análisis, visualización, consulta y combinación de la información geográfica de la universidad.
- ◆ IDEUCI incorpora la interoperabilidad entre los sistemas, la integralidad de los datos geográficos y la gestión de servicios web.

RECOMENDACIONES

Luego de la realización la aplicación IDEUCI se considera que es recomendable:

- ◆ Realizar otras versiones que integren los servicios nomenclátor y de cobertura.
- ◆ Incrementar funcionalidades de cálculo y obtención de información geográfica al visor.

BIBLIOGRAFÍA

1. **Consultores.** Revista vinculado. *Revista vinculado*. [Online] 2003. vinculando.org/beta/plataforma-de-servicios-de-mapas.html.
2. **Rodríguez, Pascual.** *UTILIDAD Y SIGNIFICADO DE LA INFRAESTRUCTURA DE DATOS ESPACIALES DE ESPAÑA*. s.l. : Instituto Geográfico Nacional, 2010.
3. **Cuba, Portal Geoespacial de la República de.** Portal Geoespacial de la República de Cuba. [Online] 2010. <http://www.iderc.co.cu/>.
4. **ProDevelop | Integración de Tecnologías.** ProDevelop | Integración de Tecnologías. [Online] 2009. <http://www.prodevelop.es/es/tecs/geo/servidoresmapas>.
5. **Gianfelici, Esteban.** Mapas y mapas.com.ar. [Online] 07 2008. [Cited: 11 10, 2011.] <http://www.mapasymapas.com.ar/el%20dato%20geografico.php>.
6. **Unidad1_2010.** *Introducción a las IDEs y los Geoservicios*. 2010.
7. **Najar.** *Qué son los metadatos*. 2006.
8. **Real Academia Española.** Diccionario de la Real Academia Española. [Online] 2005. [http://buscon.rae.es/drae/ ..](http://buscon.rae.es/drae/)
9. **World Wide Web Consortium.** World Wide Web Consortium. [Online] 2004. <http://w3c.es/Divulgacion/GuiasBreves/ServiciosWeb..>
10. **Cordón, Nuria.** ComputerWord. [Online] 2011. <http://www.idg.es/computerworld/Los-GIS-se-cuelan-en-nuestras-vidas/seccion-tecnolog%C3%ADAs/articulo-203057>.
11. **IDEE.** Infraestructura de Datos Espaciales de España. [Online] 2006. www.idee.es.
12. **SOCIALES, REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS.** *DEFINICIÓN Y DESARROLLO ACTUAL EN LA IDEE*. 2004.
13. **Nacional, Instituto Geográfico.** Infraestructura de Datos Espaciales de la República de Cuba. *Infraestructura de Datos Espaciales de la República de Cuba*. [Online] 2004. [http://www.iderc.co.cu/..](http://www.iderc.co.cu/)
14. **Canaria, Infraestructura de Datos Espaciales.** IDE Canarias (Infraestructura de Datos Espaciales de Canarias). [Online] 2011. [Cited: 11 10, 2011.] <http://www.idecan.grafcan.es/idecan/es/portal/documentacion/18-el-ogc.html>.
15. **OGC.** Open Geospatial Consortium. [Online] 2012. <http://www.opengeospatial.org/standards/gml>.
16. **Open Geospatial Consortium Inc. (OGC).** *OpenGIS® Web Map Server Implementation Specification*. s.l. : Open Geospatial Consortium Inc., 2006.
17. —. *Web Feature Service Implementation Specification*. s.l. : Open Geospatial Consortium Inc., 2005.
18. —. *Web Coverage Service (WCS), Version 1.0.0*. s.l. : Open Geospatial Consortium Inc., 2003.

19. **Open Geospatial Consortium Inc. (OGC).** *Gazetteer Service Specification*. s.l. : Open Geospatial Consortium Inc, 2001.
20. **Open Geospatial Consortium Inc. (OGC).** *OpenGIS® Catalogue Services Specification*. s.l. : Open Geospatial Consortium Inc., 2007.
21. **INSPIRE.** INSPIRE. *INSPIRE*. [Online] 2011. <http://inspire.jrc.ec.europa.eu>.
22. **IDERC.** Instituto Geográfico Nacional de la República de Cuba . [Online] 2004. <http://www.iderc.co.cu/>.
23. **Jacobson, Ivar.** *Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
24. **FLORES, MIRIAN MILAGROS DÍAZ.** *RUP vs XP. RUP vs XP*.
25. **Mora, Francisco.** *Uml. Uml*. 2003.
26. **The PHP Group.** PHP. [Online] 2012. <http://us3.php.net/manual/en/features.commandline.introduction.php>.
27. **HTML.net.** HTML.net. [Online] 2012. <http://es.html.net/tutorials/html/lesson2.php>.
28. **flash-vs-javascript-ventajas-y-desventajas.** newemage. *newemage*. [Online] 2010. <http://www.newemage.com.mx/blog/disenio-grafico/flash-vs-javascript-ventajas-y-desventajas>.
29. *Arquitectura Cliente/Servidor. Server, Microsoft Windows*. 2009, Introducción a Redes.
30. **Openlayers.** Openlayers. *Openlayers*. [Online] 2006. <http://openlayers.org/>.
31. **ibm.** blogger. *blogger*. [Online] 2007. <http://www.ibm.com/developerworks/ssa/library/x-blogger/>.
32. **mapserver.** mapserver. *mapserver*. [Online] <http://mapserver.org/>.
33. **Pressman, Roger.** *Ingeniería de Software, un enfoque práctico. Quinta edición*. s.l. : McGraw-Hill Companies, 2002.
34. **Drupal.** [Online] <http://drupal.org/es/drupal..>
35. **drupal.** drupal . *drupal* . [Online] 2009. <http://www.clarcat.com/es/productos/drupal> 2009.
36. **Pressman, Roger.** *El Proceso Unificado de Desarrollo*.
37. —. *Pressman, 6ta edición*. s.l. : McGraw-Hill Interamericana.
38. **poo-y-patrones-de-diseno-en-drupal.** poo-y-patrones-de-diseno-en-drupal. *poo-y-patrones-de-diseno-en-drupal*. [Online] 2010. <http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.

GLOSARIO

Navegador web: Es una aplicación que proporciona la visualización de información, documentos de texto, datos, animaciones, videos, hipervínculos, enlaces sobre la web. El navegador interpreta el código, generalmente escrito en HTML y lo descodifica con el objetivo de mostrarlo en pantalla y posibilita un contacto entre el usuario y el contenido.

Cliente: Es el componente que autoriza que el usuario realice un requerimiento a través de una interfaz gráfica que es transferido posteriormente al servidor. Un requerimiento puede convertirse en varias peticiones. La ubicación de los datos de la aplicación y el proceso de gestión de los estos es invisible para el cliente.

Servidor: Es un ordenador que provee los datos solicitados por parte de los navegadores de otras computadoras. Los servidores almacenan información en forma de páginas web y a través del protocolo HTTP lo entregan a petición de los clientes (navegadores web) en formato HTML.