

# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



## Sistema de Información Geográfica para la supervisión del consumo energético en la UCI

### TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

AUTOR: David Espinosa Hidalgo

TUTOR: Ing. Karel de la Cruz Rigueiro

CO-TUTOR: Ing. Amado Espinosa Hidalgo

La Habana, Junio 28 de 2011

“Año 53 de la Revolución.”

# Dedicatoria

---

*A abu:*

*Lo mejor que me ha pasado.*

*A mi hermano:*

*La estrella más radiante,*

*siempre a ocho años de distancia.*

*Te regalo el crédito de mis primeros pasos en la profesión,*

*y en la vida...*

*Y a Ustedes también:*

*Los quiero con la vida,*

*Y palabras que no puedo escribir.*

# Agradecimientos

---

*Para agradecerles a Ustedes necesito cien mil páginas más.*

*Aquí no imprimen tantas, así que, resumiendo:*

*Gracias. Por todo el amor que les debo. Por la vida.*

*Por estar a mi lado aquellos días; por llorar conmigo.*

*Por las risas, que nunca faltaron.*

*También... por supuesto... gracias por todo.*

*A Dios, o quien haya sido, por escucharme.*

*A mis amigos (casi hermanos), por el apoyo, por estar cuando los necesito; y cuando no también.*

*A Grey, por soportarme... y aún peor: por quererme.*

*A todos los que tiraron un granito de arena en la mezcla.*

*A Adrián, Alain y Karel, por los cubos de arena.*

*A mis compañeros de grupo en estos cinco años, por aceptarme y por los buenos ratos...*

*Si se queda alguien que levante la mano...*

*Ah, a ti también. Gracias.*

## Declaración de autoría

---

Declaramos que somos los únicos autores del trabajo titulado: “Sistema de Información Geográfica para la supervisión del consumo energético en la UCI”. Y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo – con carácter exclusivo – para su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

David Espinosa Hidalgo

Autor

---

Ing. Karel de la Cruz Rigueiro

Tutor

## Opinión del Tutor

---

## Opinión del Oponente

---

RESUMEN.....	IX
INTRODUCCIÓN.....	1
CAPÍTULO 1. Fundamentación teórica de la solución .....	8
1.1. Introducción.....	8
1.2. Conceptos asociados al dominio del sistema .....	8
1.3. Antecedentes y estado actual de los SIG en el mundo.....	10
1.3.1. Tipos de SIG .....	11
1.4. Principales tecnologías .....	13
1.5. Principales soluciones SIG en el mundo para la supervisión energética .....	14
1.5.1. SIG para la GDE de San Miguel de Tucumán, Argentina. Empresa EDET SA .....	14
1.5.2. Telvent GIS para la GDE. Empresa Telvent SA .....	15
1.5.3. Simulación de Redes Eléctricas Integradas a un SIG. Empresa UTE .....	16
1.5.4. Certa GIS .....	16
1.6. Conclusiones parciales .....	17
CAPÍTULO 2. Tecnologías y herramientas .....	18
2.1. Introducción.....	18
2.2. Sistema Operativo.....	18
2.3. Servidor web .....	19
2.3.1. HTTP Apache.....	19
2.4. Lenguajes de programación .....	20
2.4.1. Lenguajes de programación del lado del cliente.....	20
2.4.2. Lenguaje de programación del lado del servidor: PHP .....	21
2.5. Sistema gestor de bases de datos PostgreSQL .....	22
2.5.1. PostGIS.....	23
2.6. Servidor de mapas MapServer .....	23
2.7. Metodología de desarrollo del software .....	24
2.7.1. RUP .....	24

2.8.	Lenguaje unificado de modelado (UML) .....	26
2.9.	CartoWeb .....	27
2.10.	ExtJs.....	27
2.11.	Ajax .....	28
2.12.	Herramienta CASE .....	29
2.12.1.	Visual Paradigm .....	30
2.13.	Entorno integrado de desarrollo (IDE): Aptana .....	30
2.14.	Conclusiones parciales .....	31
CAPÍTULO 3 Presentación de la solución.....		32
3.1.	Introducción.....	32
3.2.	Entorno donde trabajará el sistema.....	32
3.2.1.	Conceptos principales del entorno .....	32
3.2.2.	Eventos principales del entorno .....	33
3.2.3.	Diagrama de clases del Modelo de Dominio .....	34
3.2.4.	Glosario de Términos del Dominio .....	34
3.3.	Requisitos funcionales .....	35
3.3.1.	RF 1. Navegar a través del mapa.....	35
3.3.2.	RF 2. Localizar contadores de consumo .....	36
3.3.3.	RF 3. Diferenciar consumo de áreas por color .....	37
3.3.4.	RF 4. Visualizar información de consumo .....	37
3.3.5.	RF 5. Visualizar relación entre varios parámetros .....	38
3.3.6.	RF 6. Exportar información visual.....	38
3.3.7.	RF 7. Guardar perfil de usuario .....	39
3.3.8.	RF 8. Visualizar capas deseadas del mapa.....	40
3.4.	Requerimientos no funcionales .....	40
3.4.1.	Fiabilidad.....	40
3.4.2.	Eficiencia.....	41
3.4.3.	Restricciones de diseño .....	41
3.4.4.	Documentación de usuarios en línea y ayuda del sistema .....	41
3.4.5.	Requisitos de hardware.....	41
3.4.6.	Requisitos de software.....	42



3.4.7.	Licencia.....	42
3.4.8.	Requisitos legales .....	42
3.4.9.	Estándares Aplicables.....	43
3.5.	Descripción del sistema propuesto.....	43
3.5.1.	Descripción de los actores del sistema.....	43
3.5.2.	Descripción de los casos de uso del sistema .....	43
3.6.	Descripción textual de los Casos de Uso del Sistema .....	44
3.6.1.	Descripción del Caso de Uso Graficar.....	44
3.6.2.	Descripción del Caso de Uso Tematizar.....	51
3.6	Conclusiones parciales .....	58
CAPÍTULO 4. Diseño e implementación de la solución.....		59
4.1	Introducción.....	59
4.2	Modelo de diseño.....	59
4.2.1	Estructura de un plugin o componente .....	62
4.2.2	Patrones de diseño .....	62
4.2.3	Diagrama de clases del diseño .....	63
4.3	Modelo de datos.....	65
4.3.1	Diagrama Entidad-Relación.....	65
4.3.2	Diagrama de clases persistentes.....	66
4.4	Modelo de implementación.....	66
4.4.1	Modelo de despliegue .....	67
4.5	Plan de pruebas .....	68
4.5.1	Objetivos.....	68
4.5.2	Descripción .....	68
4.5.3	Resultados de las pruebas .....	69
4.6	Conclusiones parciales .....	69
CONCLUSIONES .....		70
RECOMENDACIONES .....		71
TRABAJOS CITADOS .....		72
BIBLIOGRAFÍA.....		74

La Dirección Energética de la UCI es la encargada de gestionar la asignación y consumo de energía en la universidad. El proceso de supervisión – por su grado de complejidad – hace que la toma de decisiones basada en el consumo energético resulte una actividad engorrosa. Todo esto implica que el manejo de los datos referentes al consumo de energía y el estado de los grupos electrógenos no se pueda realizar de manera automática. Es por eso que surge como propuesta de solución el Sistema de Información Geográfica para la supervisión energética de la UCI (SIG Energía), el cual permitirá la georreferenciación de la información energética de los mayores consumidores de la universidad. Presenta funcionalidades que agilizan la búsqueda de las principales áreas de consumo, así como la ubicación geográfica de las mismas sobre un mapa. El sistema posibilitará la realización de análisis temáticos a partir de la información asociada a los distintos indicadores de consumo establecidos, con el objetivo de conocer las áreas que están dentro del consumo asignado, así como las que se puedan exceder, para la toma de medidas a tiempo. El SIG Energía está desarrollado para apoyar la toma de decisiones, en aras de lograr un uso racional de la energía eléctrica consumida por la universidad.

**Palabras Clave:** SIG, UCI.

# INTRODUCCIÓN

---

Con el desarrollo de la electrónica a mediados del siglo XX surgieron nuevas tecnologías para el procesamiento de datos. Estas condujeron a la creación del primer ordenador, con lo cual ha ganado en intensidad la revolución tecnológica iniciada hace más de cincuenta años. A partir de un continuo proceso de cambio e innovación, la informática se ha ido insertando en numerosos sectores de la sociedad. Como parte de este proceso han surgido nuevas disciplinas y un sinnúmero de aplicaciones en otras ramas del saber, que propician el desarrollo de la vida tal como se le conoce.

Entre sus escenarios de aplicación se encuentra la Geoinformática, disciplina que surge de la unión entre la informática y las ciencias de la tierra. Buscando una definición más formal: "(...) la Geoinformática es el uso de las matemáticas y las técnicas informáticas para resolver problemas geográficos, normalmente creando o utilizando programas informáticos, modelos matemáticos o ambos" (Guinea de Salas, 2007).

Desde sus inicios, la Geoinformática comenzó a utilizar medios más eficaces para el tratamiento de la información geográfica, lo que bien podría significar un mapa con escasa descripción de la región representada. Pero con el incesante avance de las tecnologías y los medios de comunicación, desde entonces se han venido empleando cada vez métodos más sofisticados, tanto para la recolección de la información, como para su tratamiento y representación.

Un Sistema de Información Geográfica<sup>1</sup> (SIG, en lo adelante) es a grandes rasgos una colección de hardware, software y datos geográficos diseñados para recoger, almacenar, actualizar, manipular, analizar y reproducir datos con referencias geográficas (ETSI Topografía, Geodesia y Cartografía, 2008).

Los SIG – en sus inicios – fueron creados para satisfacer una necesidad básica de la sociedad: conocer la situación de un territorio determinado, a partir de la cual se tendría la capacidad de responder ciertas preguntas con elevado valor socio-económico con mayor exactitud: ¿cuáles son las posibles zonas a

---

<sup>1</sup> GIS por sus siglas en inglés.

poblar?, ¿dónde sembrar?, ¿qué suelos son útiles para la agricultura y cómo organizarlos? Y el resultado sería, entre otros, una información detallada del área, a la cual – con el tiempo – se le han encontrado muchas aplicaciones.

Actualmente estos sistemas representan un paso significativo hacia el futuro de la geoinformática, debido a que permiten el almacenamiento y manipulación de grandes volúmenes de información geográfica. Además, entre otras características, estos destacan el análisis de patrones, relaciones y tendencia de la información, que sirven de ayuda para la inevitable toma de decisiones en situaciones excepcionales de manera eficiente.

Cuba – inmersa en un proceso de informatización de la sociedad –, en aras de mantenerse a la altura de los adelantos científico-técnicos, también ha querido insertarse en el mercado de las tecnologías informáticas de punta; pero a pesar de todos los descubrimientos que tienen lugar cada día en el mundo, existen barreras que impiden que este proceso se efectúe de forma equitativa en todas latitudes. Principal motivo por el cual Cuba aún no forma parte de la vanguardia internacional en esta área.

No obstante, se han realizado numerosos esfuerzos para alcanzar la informatización y el crecimiento económico de la nación. Entre estos figura la construcción de centros de desarrollo de software y universidades, jugando un rol fundamental la Universidad de las Ciencias Informáticas (UCI). Esta constituye uno de los principales pilares a las puertas de la nueva era de la información pues fue creada a partir de un nuevo concepto: *Universidad Productiva*. La UCI tiene la misión de confeccionar software, además de brindar servicios informáticos partiendo de la doble vinculación docente-productiva de los profesionales en formación.

La UCI tiene retos de envergadura en varias latitudes del mundo. Para dar cumplimiento a estos compromisos, la universidad está estructurada en centros productivos. Entre ellos figura el Centro de Geoinformática y Señales Digitales (GEYSED), cuya misión consiste en desarrollar productos, servicios y soluciones informáticas en el sector de las señales digitales y la geoinformática. De esta forma, el centro contribuye a la formación integral de profesionales que respondan a las necesidades del progreso científico-técnico y socio-económico del país, en aras de encontrar un lugar en el mercado internacional.

En la actualidad la institución cuenta con una amplia gama de servicios para el manejo de los recursos internos. Además se cuenta con el producto GeneSIG, el cual es una plataforma libre para el desarrollo de SIG. Este sistema tiene la capacidad de ubicar ciertos recursos en un espacio geográfico en dependencia del negocio que lo requiera, así como editar y analizar la información geográfica. Sus funcionalidades están divididas en módulos, como se relaciona a continuación:

- Módulo de navegación: Contiene las principales funcionalidades a través de las cuales el usuario interactúa con la interfaz visual, garantizando que pueda realizar varias acciones de movimiento sobre el mapa con el objetivo de visualizar la información de su interés.
- Módulo de análisis y consulta: Ofrece herramientas que permiten realizar análisis al terreno, tales como cálculo de distancia y área. Además permite realizar consultas espaciales en un área definida por el usuario, la cual puede ser rectangular, circular, poligonal o irregular.
- Módulo de autenticación: Ofrece seguridad al sistema previendo que solamente puedan acceder a la aplicación los usuarios con permisos.
- Módulo de exportación: Permite que el usuario exporte el mapa o parte de este a un formato PDF para guardar registros.
- Módulos de localización: Engloba varios módulos de búsqueda y localización de recursos.

Generalmente, la supervisión energética se realiza basada en los niveles de información existentes, por lo cual, al hecho de ser muy bajos estos niveles responden una serie de riesgos que atentan en contra de la estabilidad de la relación consumo-plan. En la universidad, actualmente, se cuenta con varios contadores eléctricos digitales. Estos tienen dispositivos de captura de datos empotrados que se encargan de enviar los datos a una base de datos cada cierto tiempo. Empero, debido a la ausencia de un sistema que acceda a la información y permita realizar la supervisión energética de forma automática a partir de los datos recogidos, este proceso se lleva a cabo manualmente.

La Dirección Energética de la UCI es la encargada de gestionar la asignación y el consumo de energía en la universidad. Para efectuar el proceso de supervisión, se designa un especialista de esta entidad, el cual

debe recorrer cada uno de los puntos con contadores y registrar el identificador del mismo y el consumo en un papel. Al finalizar, el especialista archiva los resultados para futuras consultas.

El proceso, por resultar molesto – y por su grado de complejidad –, se realiza cada diez días, lo cual propicia que los niveles de información disponibles para la toma de decisiones sean pobres en comparación con los necesarios para una supervisión estricta del consumo de la universidad. Esta situación se traduce en que, hoy por hoy, la aplicación de medidas basadas en el consumo energético resulta una actividad engorrosa. Asimismo, esta ineficiencia en el proceso de supervisión – dada por la carencia de información – ha propiciado el surgimiento de una brecha a la inestabilidad del consumo de energía interno.

Además, la existencia de fallas del fluido eléctrico – consecuencia del escenario descrito – podría devenir en efectos negativos para el proceso de producción de software y formación profesional. Esta problemática se desprende de la ausencia de un sistema de supervisión que permita el manejo de forma automática de los datos referentes al consumo de energía y el estado de los grupos electrógenos.

Basado en la situación problemática expuesta, se formula el problema a resolver de la siguiente forma:

*¿Cómo optimizar los procesos de supervisión energética existentes en la UCI y proporcionar los niveles de información necesarios para la toma de decisiones eficiente?*

El objeto de estudio de la presente investigación lo constituyen los SIG, enmarcando el campo de acción – o aspecto en el cual se centrará la investigación – en los SIG de supervisión energética.

Para orientar la investigación se cuenta con el siguiente objetivo general:

Desarrollar un SIG para la supervisión del consumo energético en la UCI. Siendo necesarios para el cumplimiento del objetivo general los siguientes objetivos específicos:

1. Elaborar estado del arte de los SIG de supervisión energética.
2. Selección y argumentación de las herramientas y tecnologías.
3. Desarrollo de la aplicación según el cronograma.

Necesitando además para dar cumplimiento a cada uno de los objetivos específicos propuestos, el desarrollo de las siguientes tareas de investigación:

1. Identificar los logros y limitaciones en los enfoques existentes sobre el desarrollo de Sistemas de Información Geográfica.
2. Identificar principales aplicaciones de supervisión energética basadas en Sistemas de Información Geográfica, así como posibles funcionalidades del sistema a partir de los elementos encontrados.
3. Seleccionar y argumentar la metodología de desarrollo de software a usar en el proceso.
4. Identificar las funcionalidades que debe brindar el sistema.
5. Diseñar el sistema.
6. Implementar el sistema.
7. Realizar las pruebas del sistema.

Teniendo, al finalizar el proceso, como posibles resultados:

1. Un SIG para el control del consumo energético en la UCI.
2. La documentación técnica asociada al proceso de desarrollo.

Para garantizar la calidad de la investigación se aplicarán los siguientes métodos científicos:

**Métodos Teóricos:**

- Histórico-Lógico: Este método será utilizado para realizar un análisis histórico de la evolución y desarrollo de los SIG en general, y particularmente de los SIG de supervisión energética, permitiendo una mejor comprensión de los mismos.

➤ Analítico Sintético: Este método será necesario para analizar y comprender la teoría y documentación relacionada con el tema de investigación. Además permitirá extraer los elementos más afines al objeto de estudio.

### **Métodos Empíricos:**

➤ Encuestas: Este método permitirá desarrollar encuestas a directivos y trabajadores asociados a los procesos de informatización y control energético de la universidad, respecto a la toma de decisiones sobre el consumo de energía en la UCI.

Se defiende la idea de que mediante el desarrollo de un SIG que permita visualizar la información existente en las bases de datos de consumo, se logrará optimizar los procesos de supervisión existentes en la actualidad; además de que proporcionaría los niveles de información necesarios para la toma de decisiones eficiente.

### **Capítulos de la investigación**

El presente documento se divide en 4 capítulos, de los cuales se expone una breve descripción a continuación.

#### **Capítulo 1:**

En el primer capítulo se abordarán los elementos relacionados con la fundamentación teórica que sustenta la presente investigación. Asimismo se hará un estudio de las principales tendencias y soluciones en el marco del objeto de estudio y se especificarán algunos conceptos asociados a la investigación.

#### **Capítulo 2:**

Serán tratados los temas concernientes a las tecnologías necesarias para la implementación del sistema. Así como una descripción de cada una teniendo en cuenta sus principales características. Además contiene una fundamentación de la selección de las herramientas que se utilizarán para dar cumplimiento a los objetivos propuestos.

#### **Capítulo 3:**



Este capítulo constituye la presentación de la solución a partir de la introducción del modelo de dominio. Contiene una relación de los principales conceptos de los elementos del sistema, además de la especificación de los requisitos y las descripciones de los casos de uso del sistema.

#### **Capítulo 4:**

Trata los temas referentes a la construcción de la solución a partir de una breve descripción del sistema a través de los principales elementos del diseño e implementación. Contiene además el modelo de datos y el plan de pruebas.

# CAPÍTULO 1. Fundamentación teórica de la solución

---

## 1.1. Introducción

En este capítulo se abordan los temas correspondientes a la fundamentación teórica de la investigación. Se analizarán los principales conceptos y tendencias actuales relacionadas con los Sistemas de Información Geográfica (SIG, en lo adelante). Se estudiarán además las principales aplicaciones y tendencias existentes en el mundo desde la perspectiva del objeto de estudio.

## 1.2. Conceptos asociados al dominio del sistema

Sistema: (Del lat. systēma, y este del gr. σύστημα). Conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto. Una combinación de componentes interactivos o interdependientes, montados para cumplir una o más funciones. Según la IEEE "Sistema es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro". Dicho de otra forma: un sistema es la interrelación armónica de un grupo de elementos o componentes, cuya individualidad cumple un fin determinado que pudiera resultar insignificante; empero como conjunto, coadyuvan en la realización de un objetivo común de mayor envergadura.

Sistema Informático: Informático es, según la RAE, relativo a la informática, o bien un conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores. De esta forma, partiendo de esta definición y la brindada previamente sobre sistema, se concluye que un sistema informático es un conjunto de recursos (informáticos o no) en armónica cohesión, cuyo fin es la automatización de información.

SIG o GIS: "Un SIG es un conjunto de herramientas basadas en un soporte informático que, manejando datos geográficos, permiten resolver problemas espaciales" (Amador Guerra, 2000).

Un SIG se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos. Los SIG son una tecnología que permite gestionar y analizar la información espacial y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato.

De las definiciones anteriores se puede extraer que la importancia de los SIG radica en que las soluciones para muchos problemas frecuentemente requieren acceso a varios tipos de información que sólo pueden ser relacionadas por geografía o distribución espacial. Sólo la tecnología SIG permite almacenar y manipular información usando geografía para analizar patrones, relaciones y tendencias en la información, todo tendiente a contribuir a tomar mejores decisiones.

Información geográfica: Se denomina información geográfica (a veces referido con el acrónimo IG) a aquellos datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales.

Objetos geográficos: Los objetos geográficos son abstracciones de elementos del mundo real que están asociadas a una posición geográfica y temporal definida, respectivamente, en un sistema de referencia espacial y temporal. Estos objetos presentes en el mundo real tienen dos tipos básicos de abstracción: ocurrencias y tipos.

Servidor de mapas: Los servidores de mapas permiten al usuario la máxima interacción con la información geográfica. Por un lado el usuario o cliente accede a la información en su formato original, de manera que es posible realizar consultas tan complejas como las que haría un SIG. Un servidor de mapas funciona enviando, a petición del cliente, desde su navegador de internet, una serie de páginas HTML (normalmente de contenido dinámico DHTML), con una cartografía asociada en formato de imagen (por ejemplo, una imagen GIF o JPG sensitiva). Un servidor de mapas es, de hecho, un SIG a través de internet.

Geoprocesamiento: Manipulación y análisis de la información con referencia geográfica.

Georreferencia: Las coordenadas de un punto que permiten su ubicación sobre la tierra.

GPS: Un sistema que, mediante la utilización de una constelación de satélites, permite determinar la posición de cualquier punto sobre la tierra con gran precisión.

GUI (Graphical User Interface): Sigla que denomina a los sistemas operativos o aplicaciones que despliegan una interfaz (o diálogo) gráfica con el usuario mediante una serie de recursos visuales, tales como ventanas, íconos, cajas de diálogo, botones, menús desplegables, etc.

SCADA: Supervisory Control And Data Acquisition. Los SCADA son aplicaciones de software para la supervisión y control automático de la producción desde la pantalla del ordenador. Es un sistema encargado de proporcionar información del proceso a diversos usuarios: operadores, supervisores de control de calidad, supervisión y mantenimiento, y permitirles cambiar el valor de determinados parámetros, variables como temperatura, presión, abertura de válvulas, entre otros muchos, de los procesos que atienden (La Torre, Argota, & Fung, 2009).

CIS: Centralized Integrated System. Es un paquete de software para el control de procesos industriales e integración software/hardware (CIS DGroup, 2007).

### **1.3. Antecedentes y estado actual de los SIG en el mundo**

Hacia los años 50's surgieron las primeras aplicaciones software de cartografía automatizada (CAD y CAM) y las primeras bases de datos que permitían el manejo de atributos en el computador. Mediante estas aplicaciones era posible la confección de sencillos mapas. No fue sino hasta finales de los 60's que surgieron sistemas que permitían integrar las bases de datos con las figuras.

En esa misma década, Roger Tomlinson creó, junto a John Herring y Jack Dangermond, el que se considera el primer SIG de la historia. El sistema, cuyo nombre fue CGIS, estaba orientado al manejo de bosques de Canadá y su estructura consistía principalmente en polígonos.

A partir de entonces y hasta la actualidad se manejaron varios modelos de SIG:

En la década del '70 se manejó el modelo orientado a registros: cada figura tenía un registro correspondiente, pero no se podía establecer una relación entre las figuras.

Más tarde en los 80's se impuso una técnica llamada: modelo orientado a capas (o también orientado a bases de datos), la cual aún perdura en muchos estudios pues es la técnica más práctica y comercialmente distribuida. Consiste en algoritmos que permitían generar las posiciones relativas mediante topología en capas.

En el año 1985 fue creado el modelo orientado a objetos donde se considera el paisaje tal como lo es realmente: todo se conforma de partes y las partes se integran y forman objetos. En este modelo no hay manejo por capas, todo se integra (Franco, 2007) y actualmente es uno de los más utilizados para la representación de información geográfica.

### **1.3.1. Tipos de SIG**

Existen varias formas de catalogar un SIG. Estas varían dependiendo de los parámetros que se tengan en cuenta. Según la forma de representar los objetos se clasifican en:

➤ Vectoriales: El terreno es representado a partir de la asociación de vectores. Cada vector representa una parte de la frontera de dicho objeto geográfico, y se localiza mediante las coordenadas del punto inicial y el punto final. De esta manera, la información espacial se estructura como una lista de coordenadas asociada a cada objeto geográfico. Un punto se representa por una pareja de coordenadas  $[x, y]$ . Una línea como una serie ordenada de parejas de coordenadas  $[x, y]$ . Un área como una serie de coordenadas  $[x, y]$  coincidiendo su fin con el primer punto, determinando así sus límites.

➤ Ráster: El terreno se representa empleando una rejilla de celdas o "píxeles", que son rectángulos regulares de igual tamaño. La información tiene estructura matricial, donde a cada celda o píxel le corresponde un valor y una localización determinados (Domínguez, 1994). Un punto se representa por una celda, una línea como un grupo de celdas interconectadas y un área es representada por varias celdas adyacentes. Con esta representación no se considera de forma explícita los objetos geográficos sino que estos nacen de la ordenación en el espacio de los valores en la rejilla (Bosque, 1992). O sea, la representación de los objetos geográficos se encuentra implícita en los píxeles contiguos del mismo valor. Es factible si se desea representar elementos naturales, o que no posean bordes marcados; además es más eficiente en procesos de búsqueda temática por puntos en el espacio.

➤ Híbridos: Es un término relativamente nuevo. Con esta expresión se intenta describir a los SIG que son capaces de incorporar datos cuya estructura es ráster, y datos vectoriales, y conjuntamente sobre ellos realizan análisis espacial. Se trata más bien de sistemas vectoriales que incluyen utilidades básicas de análisis de estructuras ráster. Por lo general, cualquier sistema de este tipo, aunque se denomine híbrido, opera en su núcleo como ráster o vectorial.

Atendiendo al cliente los SIG se dividen en: SIG de escritorio, web y móviles.

Los SIG de escritorio permiten visualizar, editar y analizar los datos geográficos de manera local a través de un servidor de mapas. Estos a su vez se clasifican en:

➤ Visores: Permite desplegar información geográfica a través de una ventana que funciona como visor, cuya función puede ser agregar capas de información.

➤ Editores: Software orientado al tratamiento que recibe la información geográfica, que antecede al análisis, pues antes de introducir datos a un SIG es necesario prepararlos para su uso en este tipo de sistemas. Se requiere transformar datos en bruto o heredados de otros sistemas en un formato utilizable por el software SIG.

➤ De análisis: Disponen de funcionalidades de análisis espacial y modelización cartográfica de procesos.

Los SIG web, por su parte, permiten la visualización de datos y el acceso a través de la web. Proporcionan funcionalidades de navegación y consulta. A causa de la arquitectura subyacente de cliente-servidor no poseen gran capacidad de procesamiento, pero tienen la ventaja de la disponibilidad a través de la web y desde cualquier sitio.

Los SIG móviles han tenido un desarrollo vertiginoso en los últimos años a causa de la revolución en la telefonía celular y la teleinformática. Estos cuentan con una gran precisión a la hora de hacer la captura de datos debido a los componentes GPS que tienen integrados.

Otros autores han intentado catalogarlos desde perspectivas diferentes. Entre ellas se propone además una clasificación atendiendo a la estructura de las bases de datos a la hora de representar la realidad (SIG puntuales, lineales o zonales).

#### 1.4. Principales tecnologías

En la actualidad los SIG proliferan y ahora cada vez son más las aplicaciones que se encuentran, y las esferas sociales en que pueden insertarse. A continuación se listan algunos de los más conocidos y utilizados en el mundo, entre otras herramientas identificadas.

➤ ArcGIS: Consiste en un conjunto de aplicaciones centradas en el tratamiento de la información geográfica y trabajo con los SIG. Fue creado por ESRI<sup>2</sup>; en su conjunto contiene aplicaciones que facilitan la captura, edición, análisis, tratamiento, diseño, publicación e impresión de información geográfica. Estas aplicaciones se engloban en familias temáticas como ArcGIS Server, para la publicación y gestión web, o ArcGIS Móvil para la captura y gestión de información en campo. ArcGIS Desktop (como el nombre lo indica, pertenece a la familia de aplicaciones ArcGIS de escritorio) es una de las más utilizadas, conjunto con las herramientas: ArcReader, ArcMap, ArcCatalog, ArcToolbox, ArcScene y ArcGlobe, además de diversas extensiones. Es distribuida comercialmente bajo tres niveles de licencias que son, en orden creciente de funcionalidades: ArcView, ArcEditor y ArcInfo.

➤ Quantum GIS: Es un SIG de código libre multiplataforma desarrollado en C++ (con la biblioteca Qt para su GUI) que permite manejar formatos ráster y vectoriales, así como bases de datos geográficas. Admite el manejo de archivos ShapeFile (\*.shp) tanto como la extensión de PostgreSQL, PostGIS. Tiene cobertura ArcInfo, GRASS y MapInfo. Entre sus ventajas destaca la posibilidad de ser usado como GUI del GRASS, integrando la potencia de análisis del último con la interfaz amigable que este posee.

➤ GvSIG: Sistema de Información Geográfica de Software Libre. Herramienta gratuita con una interfaz *userfriendly*<sup>3</sup> para manejar información geográfica. En su núcleo maneja tanto datos ráster como vectoriales, además integra en una vista datos locales y remotos. Está orientada a usuarios finales que necesiten manejar información de carácter geográfico, sean profesionales o de administraciones públicas.

---

<sup>2</sup> **Environmental Systems Research Institute**: Instituto de Investigación de Sistemas del Medioambiente. Fue fundada por Jack Dangermond, uno de los creadores del CGIS.

<sup>3</sup> **En inglés**: Amigable para el usuario. Se refiere a la interfaz de usuario de fácil usabilidad.

➤ MapInfo: Desarrollado en los Estados Unidos por MapInfo Corporation. Permite visualizar fácilmente las relaciones entre los datos y la geografía. Su versión profesional sirve de gran ayuda a los analistas de negocios, planificadores y profesionales de GIS a adquirir más conocimiento de sus mercados, compartir información, mapas y gráficos con grandes cantidades de información; contribuyendo todo esto a una mayor eficiencia en la toma de decisiones.

➤ GeneSIG: Plataforma desarrollada por la Universidad de las Ciencias Informáticas para el desarrollo de SIG. Fue implementada con herramientas y tecnologías libres, cumpliendo además con las especificaciones *OpenGIS* que establece el Open Geospatial Consortium (OGC). Permite la visualización de la información de manera sencilla así como realizar análisis y manejo de la información geográfica. Presenta una arquitectura distribuida, que cuenta con una base cartográfica, y sobre ella un conjunto de objetos que contienen información asociada.

A partir del uso de los sistemas informáticos para el manejo de información geográfica, varios países han desarrollado proyectos con este tipo de sistema. De esta forma se han valido de las funcionalidades y servicios que estos prestan en aras del progreso y de la mejora de la calidad de vida, así como una mayor eficiencia en el manejo de recursos.

Actualmente, muchas empresas comerciales proporcionan servicios derivados de Sistemas de Información Geográfica, entre los cuales se encuentran la utilización de suelos, hidrografía y masas de agua, aspectos topográficos del terreno, entre otros.

## **1.5. Principales soluciones SIG en el mundo para la supervisión energética**

### **1.5.1. SIG para la GDE<sup>4</sup> de San Miguel de Tucumán, Argentina. Empresa EDET SA**

A partir de sistemas desarrollados en la plataforma GUPTA, los cuales gestionan los movimientos de seccionamientos, transformadores, clientes, etc. en la provincia de Tucumán, se implementaron una serie de procedimientos a través de servicios web. Dichos procedimientos permiten importar los datos requeridos desde el servidor de aplicaciones para comunicarlos con el SIG. De esta manera se logra

---

<sup>4</sup> **Gestión de Distribución Eléctrica**



actualizar en tiempo real los distintos sistemas informáticos usados por los sectores que conforman la empresa.

El SIG implementado está organizado en módulos. Los módulos se encuentran distribuidos en varias estaciones de trabajo conectadas en red; cada una con responsabilidades específicas respecto a la generación de datos y actualización del sistema. La base de datos puede ser utilizada por varios sectores o departamentos de la empresa para el análisis de ingeniería (planificación, calidad de servicio, etc.).

Todos los módulos se encuentran integrados al SIG, el cual actúa como soporte de todos los requerimientos de datos de las redes y configuración básica de la topología operativa. Los módulos integrados son:

- Módulo para el cálculo de Demanda: Permite obtener una serie de resultados de potencia y energía estimados en las redes de distribución que van desde un cliente en particular hasta las barras de alimentación de los distribuidores y alimentadores.

- Módulo de Aplicaciones Eléctricas: Permite realizar estudios eléctricos sobre redes de alta tensión (AT) y obtener resultados claves a la hora de efectuar la toma de decisiones.

### **1.5.2. Telvent GIS para la GDE. Empresa Telvent SA**

SIG para el Control de Distribución Eléctrica para la Alta, Media y Baja Tensión que posibilita un control topográfico continuo de la red completa. El sistema atiende los procesos operativos y de negocio presentes en la mayoría de los servicios: llamada de emergencia, análisis de averías, diseño de trabajo de ingeniería, automatización de campo, análisis de ingeniería, decisión, soporte e informes, mantenimiento de datos, gestión de trabajo, CIS, SCADA, conmutador móvil y gestión documental.

La solución permite introducir datos cartográficos para incrementar el grado de referencia geográfica visual así como el componente identificativo. A través de un editor autónomo se realiza el mantenimiento de los datos relacionados con cada elemento de los mapas de redes. Incluye diagramas unifilares de redes de media y alta tensión; esquemas internos de subestaciones, transformadores, de tal modo que se puedan incorporar componentes adicionales.

Los datos proporcionados por el SCADA en tiempo real y las tasas de consumo del CIS, se convierten automáticamente en consumo de energía. Además, los datos obtenidos pueden ser representados gráficamente en tensiones, corrientes, flujos de energía y pérdidas. Además, se resaltan los componentes que hayan sobrepasado predeterminados límites operativos admisibles.

### **1.5.3. Simulación de Redes Eléctricas Integradas a un SIG. Empresa UTE<sup>5</sup>**

Herramienta informática corporativa desarrollada por la empresa eléctrica uruguaya UTE, mediante la cual, haciendo uso de la información disponible en un SIG, realiza cálculos eléctricos tales como flujos de carga, cortocircuitos y cálculos referentes a calidad de servicio y cuyos resultados se pueden consultar en forma de informes o en forma gráfica directamente sobre el SIG e incluso exportarlos a otras aplicaciones tales como Excel o Word. Esta herramienta corporativa es usada en todo el país por las unidades especializadas de UTE para planificar y proyectar las obras de ampliaciones y mejoras en su sistema de distribución eléctrico, así como para simular y realizar diversos cálculos eléctricos y de calidad de servicio. Combina la potencia del manejo espacial de la información geográfica, la simplicidad de una herramienta de ingeniería y la adaptabilidad de un sistema hecho a medida.

### **1.5.4. Certa GIS**

Es una completísima herramienta diseñada para incorporar las modificaciones al modelo desde el área de ingeniería de la empresa. Entre sus muchas funciones se destacan:

- Edición del modelo geográfico así como el esquema unifilar u ortogonal de la red y los mundos internos de elementos complejos, tales como subestaciones o estaciones de transformación en distribuidoras eléctricas, o cableados internos de edificios en comunicaciones.
- Incorpora herramientas automáticas de dibujo configurables y objetos de red inteligentes que hacen prácticamente imposible entrar errores durante el dibujo de redes.
- Permite realizar análisis complejos sobre los datos asociados al modelo.

---

<sup>5</sup> **Empresa Eléctrica de Uruguay**

- Corre dentro de AutoCAD teniendo acceso a todas las funciones de edición del mismo.

Además Certa Mobile permite visualizar el modelo desde un dispositivo que corra Windows Mobile o Palm OS. Cada dispositivo se configura para descargar al mismo las modificaciones del modelo de red dentro de un polígono definido en el maestro de áreas, en el momento de sincronizar el mismo.

Las principales funciones de Certa Mobile son:

- Navegación geográfica del modelo.
- Búsquedas por calle, dirección o sobre los datos que se hayan descargado al dispositivo.
- Búsquedas por los datos asociados a la red que se hayan bajado al dispositivo móvil.
- Edición de los datos asociados al modelo.
- Capacidad de realizar anotaciones gráficas (redlining).

## **1.6. Conclusiones parciales**

Luego de realizado un estudio de las aplicaciones encontradas se han podido determinar una serie de características que, si bien no tienen en común todas las aplicaciones, servirán de base para el desarrollo de la solución. Entre las funcionalidades en potencia destacan – extraídas del CertaGIS – la navegación geográfica del modelo, búsqueda por calle, dirección u otros, o por los datos asociados a la red. Del sistema de la empresa UTE se seleccionó como posible característica a agregar en la solución la realización de consultas sobre los datos en forma gráfica directamente sobre el SIG e incluso exportarlos a otras aplicaciones tales como Excel o Word. Se considera la realización de un SIG basado en tecnologías web; usando como punto de partida la plataforma GeneSIG.

# CAPÍTULO 2. Tecnologías y herramientas

---

## 2.1. Introducción

Cuba, actualmente, se encuentra en medio de un proceso de informatización que abarca los principales sectores de la sociedad. Como se ha dicho anteriormente, la UCI juega un papel fundamental dentro del mismo debido a que en ella es posible encontrar tecnología de punta a nivel internacional; además del hecho de que entre sus filas militan muchos de los profesionales que llevarán a cabo la tarea de informatizar la sociedad.

Teniendo en cuenta las necesidades del país, la universidad ha encontrado en el software libre una alternativa para la producción y comercialización de los productos desarrollados en la entidad. Además, debido al auge que ha cobrado este movimiento a nivel internacional, cada vez se hace mayor énfasis en la utilización de herramientas libres para el desarrollo de aplicaciones. En este capítulo serán tratados los temas concernientes a las tecnologías necesarias para la implementación del sistema. Es preciso tener en cuenta el significado y valor que les confiere a las mismas el tipo de licencia bajo la cual se encuentran.

## 2.2. Sistema Operativo

En la actualidad se tiene la facilidad de encontrar disímiles aplicaciones en el mundo. Existe una amplia gama de ofertas en lo que respecta al software, entre los cuales destacan, por su importancia, los sistemas operativos. A partir de que esta oferta surge en un mundo capitalista, el costo representa un factor fundamental a la hora de adquirir estos sistemas.

En su mayoría, al ser privativos o propietarios, representan un dilema ético para quien lo posee, además de que su comercialización y uso está vedada o requiere de una licencia, la cual constituye gastos extras. Por lo tanto, el simple hecho de desarrollar software en (o para) un sistema operativo de este tipo se convertiría en una cadena interminable de trabas que no todos pueden sortear con éxito y bolsillos saludables.

Es entonces cuando hace su entrada Linux como alternativa libre para el desarrollo. Este es un sistema operativo compatible con Unix, y posee dos características que lo diferencian del resto de los que existen actualmente en el mercado. Su principal ventaja: es libre, lo cual se traduce en cero gastos en licencias por su uso; además, el sistema viene acompañado del código fuente y está formado por el núcleo<sup>6</sup>, enriquecido por una enorme cantidad de bibliotecas y programas que hacen posible su utilización.

Entre otras características vale destacar que posee muchas de las características de los mejores y más usados sistemas que actualmente pueblan el mercado; entre las cuales encontramos que es multitarea, multiusuario, multiplataforma, además de que posee protección de la memoria entre procesos para evitar que alguno pueda consumir más de la necesaria.

### **2.3. Servidor web**

Los servidores web son programas que implementan el protocolo HTTP y se encargan de transferir hipertextos y páginas web. Al ejecutarse en un ordenador se mantiene esperando peticiones por parte de un cliente, una vez recibidas determina si debe enviar código estático HTML o ejecutar algún tipo de aplicación. Luego procede a ejecutar los scripts correspondientes y, a partir de este paso, generan un código HTML, que será enviado al cliente mediante el protocolo de comunicación. Una vez enviada la respuesta el cliente se encarga de procesarla para mostrar la página o algún mensaje en caso de producirse un error.

#### **2.3.1. HTTP Apache**

Es un servidor web flexible, rápido, eficiente, y continuamente actualizado, cuya principal característica reside en que su código es abierto; es multiplataforma, siendo soportado por Unix, Microsoft Windows, MacOS, entre otras. Implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Su modularidad permite que pueda ser adaptado a diferentes entornos y necesidades, con los disímiles módulos de apoyo que proporciona. Gracias a su modularidad se han desarrollado diversas extensiones entre las que destaca PHP como lenguaje de programación del lado del servidor. Presenta otras características altamente configurables, entre las que destacan bases de datos de autenticación y negociado de contenido.

---

<sup>6</sup> También conocido por el término del idioma inglés: *kernel*.

Actualmente es el servidor HTTP más usado. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente.

## **2.4. Lenguajes de programación**

Un lenguaje de programación es un idioma artificial diseñado para expresar cálculos y operaciones que pueden ser realizadas por las computadoras. Son usados para crear programas o software de diversos tipos, los cuales varían según las características del lenguaje. Están formados por una sintaxis y semántica, que son un conjunto de símbolos y reglas que definen su estructura y el significado de sus expresiones. Facilitan la tarea de programación de computadores, pues disponen de un lenguaje natural que permite que las instrucciones sean leídas y escritas por personas, a su vez resultan independientes del modelo de computador que será utilizado.

Estos pueden ser compilados e interpretados; los primeros constan de dos fases: primeramente se traduce el programa simbólico a código máquina, y luego se realiza la ejecución y el procesamiento de los datos. Por otra parte los lenguajes interpretados utilizan un programa intérprete o traductor, el cual simula una máquina virtual, y realiza la tarea de analizar la descripción simbólica del programa fuente y la ejecución de las instrucciones dadas.

### **2.4.1. Lenguajes de programación del lado del cliente**

#### **2.4.1.1. Javascript**

Javascript es un lenguaje interpretado de fácil uso, diseñado para crear aplicaciones dinámicas en la web que enlazan objetos y recursos tanto en el cliente como el servidor. Está concebido para su uso por creadores de páginas HTML y aplicaciones de empresas para programar de forma dinámica el comportamiento de los objetos existentes tanto en el cliente como en el servidor (Easton, 2001). Además no requiere de la instalación de ningún software para su visualización, salvo el intérprete de un navegador web.

Sus conceptos y diseño representan la nueva generación de software para el Internet. Sus principales características son:

- Diseñado para crear aplicaciones centradas en la red<sup>7</sup>.
- Complementario e integrado a Java.
- Complementario e integrado a HTML.
- Multiplataforma y libre.

#### **2.4.1.2. Hypertext Markup Language (Html)**

Es una aplicación del SGML<sup>8</sup> para definir tipos de documentos estructurados. Lenguaje de marcas de texto utilizado normalmente en la World Wide Web<sup>9</sup>. Su uso permite describir la estructura y el contenido de una página en forma de texto, así como complementarlo con otros objetos, que pueden ser imágenes, videos, etc. Se describe a través de etiquetas (tags) y además puede añadir scripts de otros lenguajes - como es el caso de Javascript- con el objetivo de alterar la apariencia de un documento.

#### **2.4.2. Lenguaje de programación del lado del servidor: PHP<sup>10</sup>**

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos (Puertas, 2002).

Al igual que lenguajes como ASP o JSP, se ejecuta en el servidor, de manera que el cliente sólo recibe el resultado de la ejecución en el servidor. De esta forma el cliente no puede conocer el código que ha generado esa salida. Este funcionamiento es muy diferente a lenguajes como Javascript, que se ejecutan en el cliente, pudiendo ver el usuario todo el código y su funcionamiento.

---

<sup>7</sup> **Network-centric Applications.**

<sup>8</sup> **Estándar de Lenguaje de Marcas Generalizado o Standard Generalized Markup Language por sus siglas en inglés.**

<sup>9</sup> **Es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.**

<sup>10</sup> **Hypertext Preprocessor.**

PHP ha tomado muchas de las mejores cualidades de otros lenguajes existentes: la versatilidad del C, los objetos de Java y la facilidad y potencia del analizador de Perl. Posee licencia de código abierto, lo cual implica que el código fuente es libre de ser descargado y completamente inspeccionado, y funciona en varias plataformas, Linux, BSD, y todas las versiones de Windows 9x, 2000 y NT, entre otras tantas.

Es desarrollado para dar la mayor versatilidad y flexibilidad a los usuarios que lo utilizan. Razón por la cual existen muchas extensiones del lenguaje que permiten utilizar nuevas bases de datos, protocolos, enlaces a bibliotecas, etcétera.

## 2.5. Sistema gestor de bases de datos PostgreSQL

Sistema gestor de base de datos objeto-relacional de código abierto. Permite la realización de consultas complejas, y la implementación de *triggers*<sup>11</sup>. Además ofrece la posibilidad de ser usado, modificado y distribuido libremente.

Entre sus más importantes características destacan (Hernandez & Sanchez, 2009):

- DBMS<sup>12</sup> Objeto-Relacional: Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.
- Altamente Extensible: Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Soporte SQL Comprensivo: Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones<sup>13</sup> SQL92.

---

<sup>11</sup> **Disparadores.**

<sup>12</sup> **Database Management System por sus siglas en inglés. Sistema gestor de bases de datos.**

<sup>13</sup> **JOIN por su significado en inglés. Son sentencias que permiten realizar consultas sobre la unión de varias tablas por un campo.**



### **2.5.1. PostGIS**

Es un módulo que proporciona a PostgreSQL la capacidad de almacenar, editar y analizar información de carácter geoespacial (Salinas, 2007). Es una extensión a PostgreSQL con tipos y primitivas geométricas que permite manipular información georreferenciada como un elemento más de la base de datos. Además agrega soporte para objetos geográficos al motor de base de datos objeto-relacional PostgreSQL, permitiéndole ser usado como PostGIS.

### **2.6. Servidor de mapas MapServer**

MapServer es principalmente una herramienta de visualización y mapeo; su funcionamiento hace posible que los usuarios puedan acceder a la información a través de un navegador web u otros protocolos de comunicación e intercambio de datos por internet. Esto permite compartir información visual sobre mapas y el acceso por parte de otras aplicaciones usando especificaciones OGC<sup>14</sup>.

Es una de las pocas soluciones al alcance que soportan distintos formatos de datos, pues mediante el uso de las bibliotecas GDA/OGR puede acceder a varios de estos formatos sin realizar conversiones de datos (Mitchell, 2005). De esta forma, MapServer puede realizar pseudo-conversiones de datos al leer varios formatos y proveer de acceso a otras aplicaciones servidores mediante un protocolo estándar.

Aunque no es una herramienta de análisis, puede representar información geográfica usando técnicas cartográficas para visualizar los resultados. Se puede encontrar en varios sistemas operativos al igual que las herramientas que le brindan soporte.

Entre otras ventajas, presenta funcionalidades que pueden ser accedidas desde una gran variedad de lenguajes de programación haciendo posible que sea integrado a partir de sus funcionalidades y usado por programas desarrollados por un usuario. Al ser de código abierto, los desarrolladores pueden mejorar, arreglar y personalizar el código subyacente al MapServer para diversos usos.

---

<sup>14</sup> Open Geospacial Consortium.

## **2.7. Metodología de desarrollo del software**

La tendencia de crear sistemas más sofisticados, adaptados a las nuevas tecnologías, a las necesidades de los usuarios que cambian constantemente, de mejorar los productos de una versión a otra, y de realizar todo este proceso de una forma más rápida, hace cada vez más complejo el proceso de desarrollo de software.

“Las metodologías de desarrollo de software definen quién está haciendo qué, cuándo y cómo para alcanzar un determinado objetivo” (Autores, 2009).

Existen varias metodologías de desarrollo de software aplicadas en los Centros de Desarrollo de la universidad, tales con RUP, XP y SCRUM. Para el desarrollo del software – y la obtención de un producto final con la calidad necesaria en el tiempo establecido – es preciso utilizar una metodología de desarrollo del software. Para la selección de la misma previamente debe realizarse un análisis de las características del proyecto con el objetivo de determinar cuál resulta más adecuada. En el presente caso, debido a la experiencia acumulada y su aplicación en el desarrollo de la plataforma GeneSIG, se utiliza RUP. Esta metodología brinda buenos resultados y un desarrollo del software efectivo; además, la misma documenta de forma detallada todo el proceso.

### **2.7.1. RUP**

RUP es más que un simple proceso para disciplinar, asignar tareas y responsabilidades. Tiene como objetivo principal, asegurar la producción de software de calidad dentro de plazos y presupuestos previsibles.

Sus características permiten que esta metodología sea adaptable a una gran variedad de sistemas para diferentes áreas de aplicación, diferentes tipos de organización y diferentes tamaños de proyecto. La particularidad de que cada ciclo de iteración exige el uso de artefactos, es el motivo que hace que sea una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

El ciclo de vida de RUP se caracteriza por:

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- **Iterativo e Incremental:** Aunque pueda parecer que los flujos de trabajo se desarrollan en cascada, RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

RUP divide su ciclo de vida en cuatro fases y nueve flujos de trabajo– seis de ingeniería y tres de soporte– como se muestra a continuación.

#### ➤ Fases de RUP

- ✓ Inicio: establece la visión y el alcance del proyecto, y las partes interesadas deben realizar la estimación de tiempo y costo.
- ✓ Elaboración: se analiza el dominio del problema para establecer una arquitectura base sólida para el desarrollo exitoso del proyecto.
- ✓ Construcción: tiene como objetivo llegar a obtener la capacidad operacional inicial, así como desarrollar y probar el producto.
- ✓ Transición: su objetivo fundamental es llegar a obtener la liberación del proyecto.

### ➤ Flujos de Trabajo de Ingeniería

- ✓ Modelado del negocio: Se realiza el entendimiento entre clientes y desarrolladores para concebir las necesidades del negocio que serán abarcadas.
- ✓ Requerimientos: Se realiza el acuerdo entre desarrolladores y clientes de lo que el sistema necesariamente debe hacer.
- ✓ Análisis y Diseño: se realiza una descripción de cómo se implementará el sistema, centrándose en la noción que se tiene de la arquitectura.
- ✓ Implementación: Se crea el software, ajustándolo a la arquitectura y asegurando que tenga el comportamiento deseado.
- ✓ Pruebas: Se realizan las pruebas que aseguran que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- ✓ Despliegue: El producto final se hace llegar a sus usuarios finales.

### ➤ Flujos de Trabajo de Soporte

- ✓ Administración de configuración y cambios: controla los artefactos producidos y las versiones del producto.
- ✓ Administración de proyectos: se manejan los riesgos y administran horarios y recursos.
- ✓ Ambiente: se brinda una guía en la configuración de un ambiente apropiado para cada proyecto.

## 2.8. Lenguaje unificado de modelado (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de visualización, especificación, construcción y documentación de los artefactos de un sistema. Ofrece un estándar para confeccionar modelos o descripciones del mismo, los cuales incluyen aspectos conceptuales, tales como procesos de negocios y

funciones del sistema, y aspectos concretos, como son expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. Está compuesto por elementos gráficos que se relacionan entre sí dando lugar a diagramas.

## **2.9. CartoWeb**

Es una aplicación comprensible y fácil de usar basado en las tecnologías web. Constituye un framework poderoso para la construcción de aplicaciones personalizadas avanzadas. Fue escrito en PHP5 basándose en el motor de MapServer. Es altamente modular y personalizable gracias a su arquitectura orientada a objetos. Es multiplataforma, de manera que puede utilizarse tanto en plataformas Windows como Unix. Puede ser configurado como un servicio Web SOAP que permite tener un servidor front-end en un ordenador, y los datos y la generación de mapa en otro. Además de que puede asociarse a PostgreSQL/PostGIS para mostrar su verdadero poder.

## **2.10. ExtJs**

ExtJs es una biblioteca de Javascript que permite construir aplicaciones complejas en Internet. Esta biblioteca incluye:

- Componentes UI<sup>15</sup> del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias de código abierto y comercial.

ExtJS es concebido como un motor que permite crear aplicaciones RIA<sup>16</sup> mediante Javascript. Una de las grandes ventajas que provee su uso es que permite la creación de aplicaciones complejas utilizando

---

<sup>15</sup> **User Interface o Interfaz de Usuario por su significado en el idioma español.**

<sup>16</sup> **Rich Internet Applications. Su traducción es Aplicaciones Ricas de Internet.**

componentes predefinidos así como un manejador de vistas o layout. Usar un motor de renderizado como ExtJS provee además otros beneficios:

- Existe un balance entre el cliente y el servidor al distribuir el procesamiento. Una vez que la carga en el servidor es más ligera, este puede manejar mayor número de clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de renderizado puede comunicarse con el servidor sin la necesidad de esperar por una acción por parte del usuario.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

## 2.11. Ajax

Ajax<sup>17</sup> – cuyas iniciales significan Javascript y XML asíncrono –, es una técnica usada en la creación de aplicaciones web interactivas. Su objetivo es eliminar la necesidad de refrescar toda la información de una página web cada vez que el usuario hace un cambio. Esto se logra mediante un intercambio de pequeñas cantidades de datos con el servidor de aplicaciones de fondo. Su resultado final se traduce en un aumento de la interactividad, la velocidad y la usabilidad. Por sí solo no es una tecnología, pero sí un término que se refiere al uso de un conjunto de tecnologías.

Esta técnica utiliza una combinación de:

- XHTML(o HTML), CSS (hojas de estilo en cascada), para maquetar y dar estilo a la información.
- DOM accedido por un lenguaje de scripting de cliente, especialmente las implementaciones de ECMAScript como Javascript o JScript, para dinámicamente mostrar e interactuar con la información expuesta.
- El objeto XMLHttpRequest para intercambiar asíncronamente información con el servidor web.

---

<sup>17</sup> **Asincronous JavaScript and XML.**

- XML, como el formato más extendido de intercambio de información entre el servidor y el cliente.

El modelo tradicional de aplicaciones web funciona enviando, a raíz de la acción de un usuario en la página, una serie de peticiones síncronas al servidor de aplicaciones. Luego, este servidor realiza una serie de procesos o lógica de negocio, y devuelve una página HTML como respuesta. Técnicamente parece tener mucho sentido, pero mientras el servidor realiza sus procesos, el usuario final está esperando y en cada interacción con la aplicación espera un poco más.

A diferencia de este modelo, las aplicaciones web que incluyen tecnología AJAX, asumen que – una vez cargado el código en la interfaz gráfica – cada interacción del usuario con la página web no implica un alto en el camino mientras la aplicación envía una petición al servidor y espera la respuesta. Por el contrario, la página web no se recarga, sino que realiza una petición asíncrona al servidor mediante XMLHttpRequest y espera la respuesta en formato normalmente XML, para – una vez recibida y procesada con DOM – redibujar la porción de la interfaz gráfica que corresponda.

La utilización de AJAX tiene asociado un conjunto de ventajas:

- Hace posible la construcción de aplicaciones web dinámicas de manera sencilla y rápida.
- Se utilizan los recursos de todas las máquinas clientes en lugar del servidor.
- AJAX permite realizar procesos en la computadora cliente (con JavaScript) con datos provenientes del servidor.
- El proceso de crear la página es realizado en el servidor antes de ser enviada a través de la red.

## **2.12. Herramienta CASE**

Son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de un software. Estas herramientas son de utilidad en cada parte del ciclo de vida del mismo pues en la planificación de tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

### 2.12.1. Visual Paradigm

Es una poderosa herramienta CASE que utiliza UML para el modelado, es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencillo de usar, fácil de instalar y actualizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Visual Paradigm ofrece:

- Un entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas.

### 2.13. Entorno integrado de desarrollo (IDE): Aptana

Aptana Studio es un IDE<sup>18</sup> gratuito, de código abierto y multiplataforma. Puede ser instalado en varios sistemas operativos: Windows, Linux o Mac OSX. Provee soporte para lenguajes como: PHP, Python, CSS, Ajax y HTML, entre otros. Da la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades.

Es ampliamente utilizado en el mundo por desarrolladores en la confección de páginas web, sobre todo para programar la parte del código que es ejecutada del lado del cliente como los scripts JavaScript.

---

<sup>18</sup> IDE. Integrated Development Environment.



Cuenta con un buen auto-completamiento por lo que contribuye notablemente a la reducción del tiempo de producción.

## **2.14. Conclusiones parciales**

Para darle solución a la situación problemática se ha decidido implementar un sistema de información geográfica que permita realizar una supervisión del consumo energético de la universidad. En el presente capítulo fue plasmada una visión global de las principales tecnologías necesarias para el desarrollo del sistema. Para esto se tuvieron en cuenta las decisiones de la dirección de arquitectura del proyecto y las políticas del mismo.

Una vez terminado este proceso, fue seleccionado Ubuntu 10.10 como sistema operativo para desarrollar la solución. Será utilizada la plataforma genérica GeneSIG (versión 1.0) como punto de partida para el desarrollo, la cual será un SIG basado en tecnologías web. El protocolo a utilizar para la creación del servicio Web y el acceso a algunas clases de CartoWeb será SOAP. Se seleccionó como servidor web la versión 2.2 de Apache y como SGBD<sup>19</sup> PostgreSQL 8.4, con su módulo PostGIS (versión 1.4) para el manejo de información georreferenciada. Como servidor de mapas se utilizará MapServer 5.4. El lenguaje de programación del lado del servidor que permitirá la implementación del sistema será PHP 5.3 y la biblioteca php-mapscrip para el manejo de objetos del mapa. La metodología que guiará el proceso de desarrollo del software será RUP, y para realizar el modelado del sistema será utilizado el lenguaje UML 2.0; haciendo uso del Visual Paradigm – en su versión 6.4 – como herramienta CASE. Para el diseño de la interfaz de usuario además se usará HTML como lenguaje de marcas complementado por Javascript y la biblioteca ExtJS, versión 3.3.0. Una vez concluido el capítulo es posible adentrarse en la propuesta del sistema.

---

<sup>19</sup> Sistema Gestor de Bases de Datos.

# CAPÍTULO 3 Presentación de la solución

---

## 3.1. Introducción

El presente capítulo corresponde a la presentación de la solución. En él se describen los elementos más significativos del dominio y los requerimientos. Se ofrecen además descripciones de la estructura y funcionamiento de la organización y el sistema como tal; y para complementar estas actividades se confeccionará también la especificación de los requisitos y las descripciones de los casos de uso del sistema.

## 3.2. Entorno donde trabajará el sistema

“Un Modelo de Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos representan las “cosas” que existen o los eventos que suceden en el entorno donde trabaja el sistema” (Jacobson, 2000). Su objetivo es conocer el funcionamiento del sistema a través de la comprensión y descripción de las clases enmarcadas dentro del contexto del mismo.

Los diagramas resultantes del modelo del dominio, especialmente el diagrama de clases, son de gran ayuda para mostrar al usuario, revisores y otros desarrolladores las clases del dominio y cómo interactúan o se relacionan mediante asociaciones.

### 3.2.1. Conceptos principales del entorno

- Cartógrafo: Persona que se dedica profesionalmente a la realización de cartas geográficas, al estudio y elaboración de mapas.
- GeoCuba: Grupo empresarial que se dedica a la elaboración, producción y venta de planos, mapas y cartas náuticas con diversos fines, así como a la realización de estudios geográficos, de impacto ambiental, e investigaciones científicas en ramas del campo de las geociencias, entregando a sus clientes, productos informativos terminados con una alta calidad y fiabilidad.

- Especialista de la Dirección Energética: Persona perteneciente a la UCI, encargada de la supervisión energética de las instalaciones de la universidad, que en algún momento necesite trabajar o consultar algún tipo de información socioeconómica o de consumo incluida en un mapa.
- UCI: La universidad es la encargada de solicitar un servicio determinado utilizando un mapa y que proporcione la información socioeconómica y de consumo referente a la misma.
- Mapa: Es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. Que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.
- Escala: Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en forma de proporción o fracción, con una escala gráfica o con una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes.
- Leyenda: Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa; suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso. Los símbolos empleados en los mapas pueden llegar a contener un gran volumen de información, que por su facilidad de lectura permiten una rápida interpretación.
- Información de consumo: Es un conjunto organizado de datos procesados referentes al consumo energético de cualquier lugar de la universidad.
- Información socioeconómica: Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés de la universidad.

### **3.2.2. Eventos principales del entorno**

Los mapas están compuestos por varias escalas representativas o leyendas, que permiten un mejor entendimiento de los mismos y la tipografía. Esta última se refiere a la variedad de mapas existentes de acuerdo con su especificación, que pueden ser de división político-administrativa, vegetación, yacimientos geológicos, entre otros. Estos mapas, además, contienen toda la información socioeconómica, tratada en los distintos tipos de datos referentes a todos los sectores de la economía y la sociedad.

La creación de los mapas que serán utilizados es responsabilidad de la entidad GeoCuba, siendo entonces la UCI quien proporcione toda la información complementaria, socioeconómica y de consumo, que le será agregada posteriormente a dichos mapas.

### 3.2.3. Diagrama de clases del Modelo de Dominio

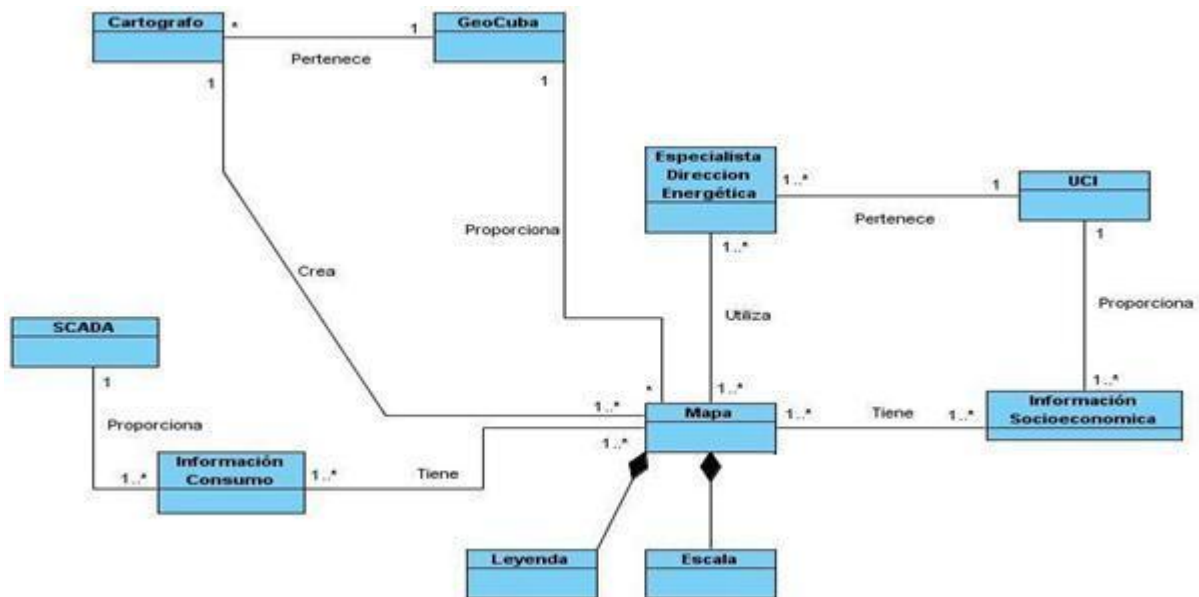


Figura 3. 1 Diagrama de clases del Modelo del Dominio

### 3.2.4. Glosario de Términos del Dominio

➤ Actor: “Los actores representan terceros fuera del sistema que colaboran con el sistema” (Jacobson, 2000). Estos pueden ser personas o sistemas informáticos, cuyo comportamiento está determinado por el rol que juegan en el sistema.

- Caso de Uso (CU): “Cada forma en que los actores usan el sistema debe ser representada por un caso de uso. Los casos de uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores” (Jacobson, 2000). Estos artefactos especifican un flujo de acciones que el sistema puede realizar para interactuar con sus actores incluyendo alternativas o flujos alternos.
- Diagrama de Casos de Uso (DCU): “Un diagrama que muestra un conjunto de casos de uso y de actores y sus relaciones (...)” (Jacobson, 2000). Estos diagramas muestran las relaciones entre actores y casos de uso desde un punto de vista estático.
- Diagrama de Clases (DC): “Muestra un conjunto de clases, interfaces y colaboraciones, y las relaciones entre éstos” (Jacobson, 2000).
- Clase: “Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica (Jacobson, 2000). Son una entidad abstracta que representa un objeto del mundo real con su comportamiento, relaciones y colaboraciones con otras clases.

### 3.3. Requisitos funcionales

Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica (Echevarría, 2004). En otras palabras, los requisitos funcionales son el núcleo de funcionamiento de un sistema, y representan un conjunto de funcionalidades que debe cumplir y muestran cómo serán ejecutados los casos de uso. A continuación se presenta la especificación de requisitos del sistema.

#### 3.3.1. RF 1. Navegar a través del mapa

<b>Nombre</b>	<b>RF 1 Navegar a través del mapa</b>
<b>Asociados</b>	<b>RF 1.1 Mover mapa</b> <b>RF 1.2 Modificar escala de visualización</b>

<b>Descripción</b>	Con este requerimiento se busca darle al especialista la posibilidad de navegar a través del mapa y visualizar desde distintas escalas el objetivo de su preferencia o una panorámica global de la zona. Asimismo, puede utilizar una forma alterna de navegación mediante el uso del mapa de referencia.
<b>Seguimiento</b>	CUS Realizar Navegación
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Crítico

### 3.3.2. RF 2. Localizar contadores de consumo

<b>Nombre</b>	RF 2 Localizar contadores de consumo
<b>Asociados</b>	No tiene.
<b>Descripción</b>	Con este requerimiento se quiere que el especialista pueda conocer rápidamente la ubicación geográfica de un dispositivo de control de consumo en el mapa.
<b>Seguimiento</b>	CUS Localizar
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Secundario

### 3.3.3. RF 3. Diferenciar consumo de áreas por color

<b>Nombre</b>	RF 3 Diferenciar consumo de áreas por color
<b>Asociados</b>	<b>RF 3.1</b> Generar consultas rápidas por color basándose en superávit <b>RF 3.2</b> Actualizar consultas automáticamente basándose en fecha actual
<b>Descripción</b>	Con este requerimiento se quiere que el usuario pueda diferenciar a simple vista el nivel de consumo de varias áreas. Dicho consumo puede ser un valor de consumo diario, semanal, o mensual.
<b>Seguimiento</b>	CUS Tematizar
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Crítico

### 3.3.4. RF 4. Visualizar información de consumo

<b>Nombre</b>	RF 4 Visualizar información de consumo
<b>Asociados</b>	No tiene.
<b>Descripción</b>	Con este requerimiento se quiere que el especialista pueda seleccionar un contador determinado y visualizar información detallada de consumo que este posee.

<b>Seguimiento</b>	CUS Ver Datos
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Secundario

### 3.3.5. RF 5. Visualizar relación entre varios parámetros

<b>Nombre</b>	RF 5 Visualizar relación entre varios parámetros
<b>Asociados</b>	RF 5.1 Generar gráficas rápidas basándose en superávit
<b>Descripción</b>	Con este requerimiento se quiere que el especialista pueda visualizar la relación que existe entre varios parámetros, tales como el consumo y el plan, de las áreas que posean dicha información. Esta será visualizada a través de gráficos que permitan percibir a simple vista una panorámica de los datos solicitados.
<b>Seguimiento</b>	CUS Graficar
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Crítico

### 3.3.6. RF 6. Exportar información visual



<b>Nombre</b>	<b>RF 6</b> Exportar información visual
<b>Asociados</b>	No tiene
<b>Descripción</b>	Con este requerimiento se quiere que el usuario pueda exportar cualquier sector del mapa en la escala deseada a un fichero en formato .PDF. Para darle formato a la selección por defecto se podrán modificar los valores correspondientes a las coordenadas, escala, formato, rótulo de la región del mapa a imprimir. Además el usuario puede definir la región que desea imprimir al seleccionar la opción Seleccionar área libre y dibujar en el mapa la misma.
<b>Seguimiento</b>	CUS Exportar
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Secundario

### 3.3.7. RF 7. Guardar perfil de usuario

<b>Nombre</b>	<b>RF 7</b> Guardar perfil de usuario
<b>Asociados</b>	No tiene
<b>Descripción</b>	Con este requerimiento se quiere que el usuario pueda realizar cambios en el objeto del mapa y poder consultarlos cada vez que acceda al sistema.
<b>Seguimiento</b>	CUS Autenticar

<b>Estado</b>	Aprobado
<b>Prioridad</b>	Crítico

### 3.3.8. RF 8. Visualizar capas deseadas del mapa

<b>Nombre</b>	RF 8 Visualizar capas deseadas del mapa
<b>Asociados</b>	No tiene
<b>Descripción</b>	Con este requerimiento se quiere que el usuario pueda realizar cambios en el objeto del mapa y visualizar las capas que desee en cualquier momento.
<b>Seguimiento</b>	CUS ControlDeCapas
<b>Estado</b>	Aprobado
<b>Prioridad</b>	Secundario

### 3.4. Requerimientos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo, estándares, etc. Los requerimientos no funcionales son aquellos que no están en directa vinculación con las funciones específicas que brinda el sistema. Se refieren a las propiedades procedentes de éste como la fiabilidad, usabilidad, eficiencia en cuanto a la respuesta en el tiempo y la capacidad de almacenamiento, etcétera.

#### 3.4.1. Fiabilidad

El sistema no debe requerir mantenimiento constante, ni optimizaciones en el almacenamiento. El sistema debe tener un tiempo promedio de 6 meses entre posibles fallas.

#### **3.4.2. Eficiencia**

El tiempo de respuesta entre una acción sobre el sistema y una respuesta del mismo no debe ser excesivo. El mismo debe oscilar entre los 20 milisegundos y 10 segundos.

#### **3.4.3. Restricciones de diseño**

- Se deben emplear los estándares establecidos (base de datos y codificación).
- Se debe lograr un producto altamente configurable y extensible, teniendo en cuenta que se desarrollará sobre la Plataforma GeneSIG y que constituye una plataforma de desarrollo para ser personalizada como aplicaciones a la medida, pudiéndose incorporar a ésta nuevas funcionalidades.

#### **3.4.4. Documentación de usuarios en línea y ayuda del sistema**

La ayuda del sistema debe estar disponible como parte del sistema.

#### **3.4.5. Requisitos de hardware**

##### **3.4.5.1. Para las PCs clientes:**

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

##### **3.4.5.2. Para los servidores:**

- Se requiere tarjeta de red.

- El Servidor de Mapas debe tener como mínimo 2GB de RAM y 2GB de disco duro.
- El Servidor de BD debe tener como mínimo 2GB de RAM y 10GB de disco duro.
- Procesador de 3 GHz como mínimo.

### **3.4.6. Requisitos de software**

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por lo tanto, deben cumplir con los siguientes requerimientos mínimos de software:

#### **3.4.6.1. Para las PCs clientes:**

- Un navegador como Mozilla Firefox, Safari u otro navegador que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux, Windows y Mac OS.

#### **3.4.6.2. Para los Servidores:**

- Sistemas operativos GNU/Linux o Windows Server 2000 o superior.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.2.2 o superior, con extensión php-mapscript.

### **3.4.7. Licencia**

Todos los componentes del sistema y las herramientas a utilizar se encuentran bajo la Generic Public License (GPL) y Lesser GPL.

### **3.4.8. Requisitos legales**

El sistema será amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

### 3.4.9. Estándares Aplicables

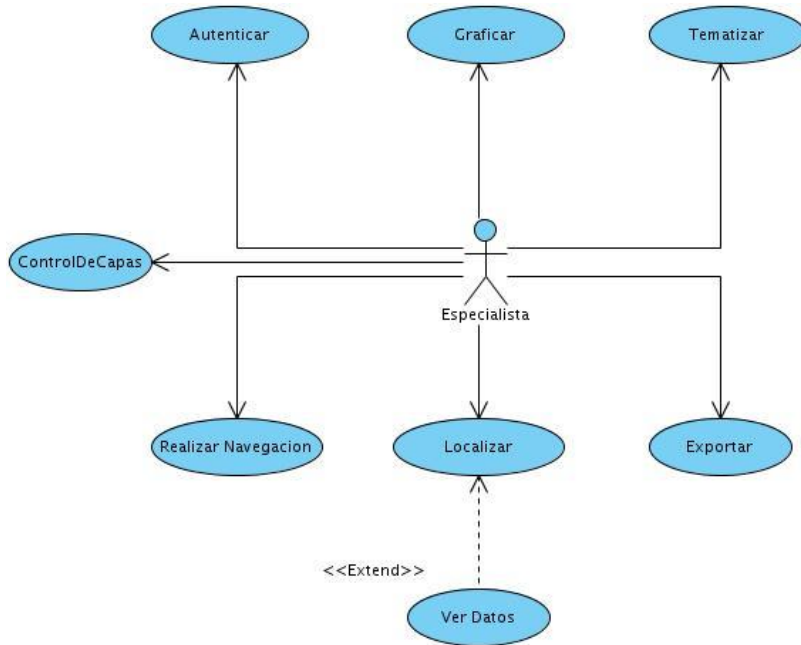
El sistema será desarrollado bajo estándares OpenGIS como aseguramiento de la parte científica y en el desarrollo se codificará y modelará siguiendo los patrones de las normativas, tanto de codificación como de diseño de bases de datos.

## 3.5. Descripción del sistema propuesto

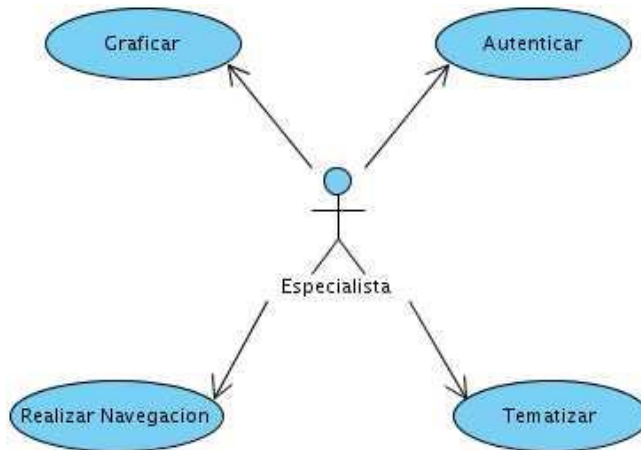
### 3.5.1. Descripción de los actores del sistema

Actor	Descripción
Especialista	Persona perteneciente a la UCI, encargada de la supervisión energética de las instalaciones de la universidad, que en algún momento necesite trabajar o consultar algún tipo de información general o de consumo incluida en un mapa.

### 3.5.2. Descripción de los casos de uso del sistema



**Figura 3. 2** Diagrama de Casos de Uso del Sistema



**Figura 3. 3.-** Diagrama de Casos de Uso Arquitectónicamente Significativos del Sistema

### 3.6. Descripción textual de los Casos de Uso del Sistema

#### 3.6.1. Descripción del Caso de Uso Graficar

<b>Caso de Uso:</b>	Graficar	
<b>Actores:</b>	Especialista	
<b>Propósito</b>	Este caso de uso se realiza con el propósito de permitir al usuario graficar los parámetros seleccionados mediante gráficos de barras y líneas.	
<b>Resumen:</b>	Este caso de uso se inicia cuando el usuario desee graficar algún indicador seleccionando la opción de crear gráficos y finaliza cuando el sistema devuelve el gráfico asociado a los parámetros seleccionados por el usuario.	
<b>Precondiciones:</b>	No tiene.	
<b>Referencias</b>	RF 5	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El usuario pulsa el botón izquierdo del mouse sobre el botón "Graficar" y selecciona una de las siguientes opciones, (ver Prototipo de Interfaz 1).</p> <ul style="list-style-type: none"> <li>➤ "Crear gráfica libre"</li> </ul>	<p>2. Si seleccionó</p> <ul style="list-style-type: none"> <li>➤ "Crear gráfica libre" ver sección "Crear gráfica Libre".</li> <li>➤ "Gráfica rápida del día anterior" ver sección "Gráfica</li> </ul>	

<ul style="list-style-type: none"> <li>➤ “Gráfica rápida del día anterior”</li> <li>➤ “Gráfica Rápida Semana Actual”</li> <li>➤ “Gráfica Rápida Mes Actual”</li> </ul>	<p>rápida del día anterior”.</p> <ul style="list-style-type: none"> <li>➤ “Gráfica Rápida Semana Actual” ver sección “Gráfica Rápida Semana Actual”.</li> <li>➤ “Gráfica Rápida Mes Actual” ver sección “Gráfica Rápida Mes Actual”.</li> </ul>
<b>Sección “Crear gráfica libre”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El usuario selecciona la opción “Crear gráfica Libre”.</p>	<p>2. El sistema muestra la ventana “Visualización de Gráfica”. Esta contiene los nombres de las principales áreas con información de consumo (A), y permite al usuario seleccionar las áreas que desea analizar; además posee un campo donde se visualizan las áreas que han sido seleccionadas previamente (B). El usuario podrá adicionar o eliminar las áreas deseadas mediante el uso de dos botones (C). La interfaz cuenta también con un campo para que el usuario seleccione el tipo de gráfica que desea crear (E) y los</p>



	<p>criterios por los cuales es posible confeccionar la gráfica: los datos disponibles y un rango de fechas a partir del cual se solicitarán los mismos (D, ver Prototipo de Interfaz 2).</p>
<p>3. El usuario llenará los campos requeridos y presiona el botón Graficar.</p>	<p>4. El sistema procesa la información y muestra la gráfica en una nueva ventana (ver Prototipo de Interfaz 3).</p>
<p><b>Flujos Alternos</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>3.2 El usuario selecciona un rango de fechas incorrecto y pulsa el botón "Graficar".</p>	<p>4.2 El sistema muestra una alerta indicando al usuario que los datos ingresados son incorrectos.</p>
<p><b>Sección "Gráfica rápida del día anterior"</b></p>	
<p><b>Flujo Normal de Eventos</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El usuario selecciona la opción</p>	<p>2. El sistema inicializa los valores que serán enviados al servidor con</p>

"Gráfica rápida del día anterior".	la fecha del día anterior al actual.
	3. El sistema muestra una ventana permitiéndole al usuario que seleccione el área que desea graficar.
3. El usuario selecciona el área deseada y presiona el botón "Graficar".	4. El sistema procesa la información y muestra la gráfica en una nueva ventana (ver Prototipo de Interfaz 3).
<b>Sección "Gráfica Rápida Semana Actual"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción "Gráfica Rápida Semana Actual".	2. El sistema inicializa los valores que serán enviados al servidor con un rango de fechas entre la fecha del día actual y la fecha de siete días atrás.
	4. El sistema muestra una ventana permitiéndole al usuario que seleccione el área que desea graficar.

<p>3. El usuario selecciona el área deseada y presiona el botón “Graficar”.</p>	<p>4. El sistema procesa la información y muestra la gráfica en una nueva ventana (ver Prototipo de Interfaz 3).</p>
<p><b>Sección “Gráfica Rápida Mes Actual”</b></p>	
<p><b>Flujo Normal de Eventos</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1. El usuario selecciona la opción “Gráfica Rápida Mes Actual”.</p>	<p>2. El sistema inicializa los valores que serán enviados al servidor con un rango de fechas entre la fecha del día actual y la fecha de treinta días atrás.</p>
	<p>5. El sistema muestra una ventana permitiéndole al usuario que seleccione el área que desea graficar.</p>
<p>3. El usuario selecciona el área deseada y presiona el botón “Graficar”.</p>	<p>4. El sistema procesa la información y muestra la gráfica en una nueva ventana (ver Prototipo de Interfaz 3).</p>
<p><b>Prototipo de Interfaz</b></p>	



Figura 3. 4 Prototipo de Interfaz 1

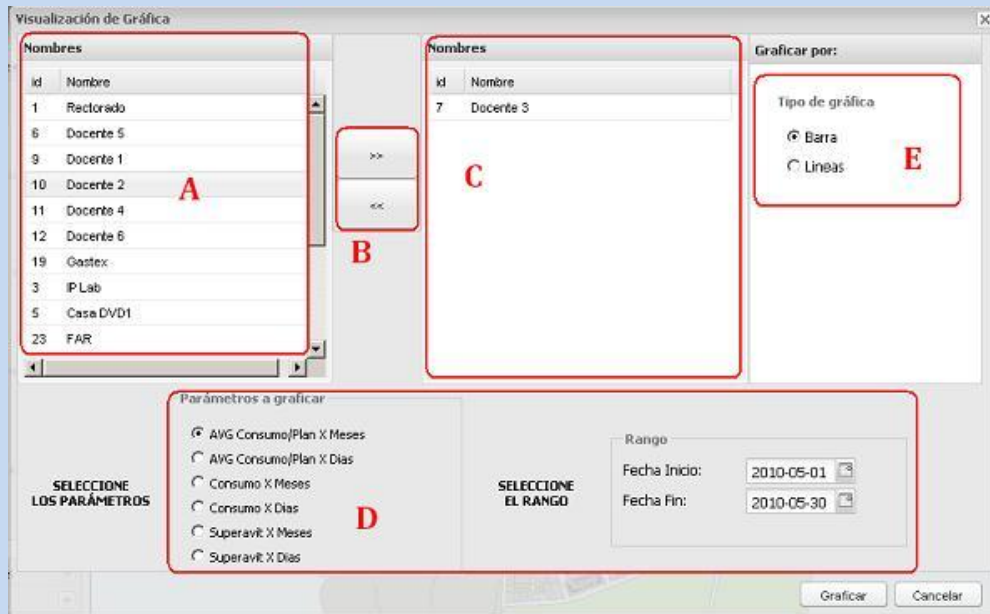


Figura 3. 5.- Prototipo de Interfaz 2

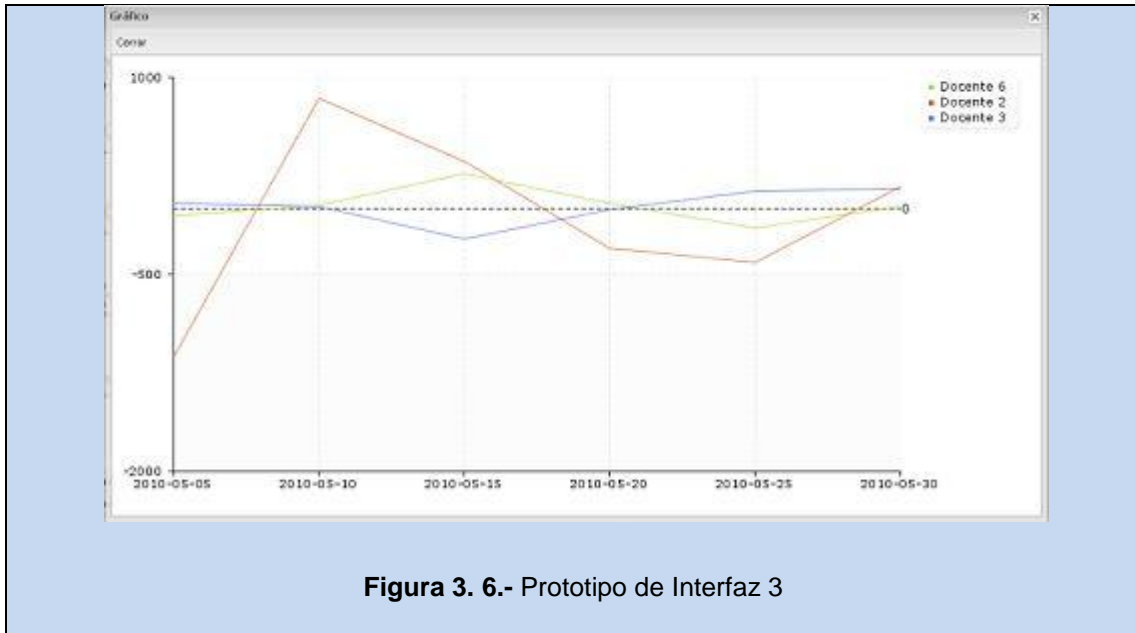


Figura 3. 6.- Prototipo de Interfaz 3

<b>Poscondiciones</b>	No tiene
-----------------------	----------

### 3.6.2. Descripción del Caso de Uso Tematizar

<b>Caso de Uso</b>	Tematizar
<b>Actores</b>	Especialista
<b>Propósito</b>	Este caso de uso se lleva a cabo con el propósito de poder visualizar un mapa temático a partir de los valores asociados a los criterios de análisis.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea realizar una acción temática sobre un área (o varias), insertando los valores correspondientes a los criterios de análisis, y termina visualizándose el mapa temático.

<b>Precondiciones</b>	No tiene
<b>Referencias</b>	RF 3
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El usuario pulsa el botón izquierdo del mouse sobre el botón “Tematizar” y selecciona una de las siguientes opciones, (ver Prototipo de Interfaz 4)</p> <ul style="list-style-type: none"> <li>➤ “Tematización Libre”</li> <li>➤ “Tematización Rápida Día Anterior”</li> <li>➤ “Tematización Rápida Semana Actual”</li> <li>➤ “Tematización Rápida Mes Actual”</li> </ul>	<p>2. Si seleccionó</p> <ul style="list-style-type: none"> <li>➤ “Tematización Libre” ver sección “Tematización Libre”.</li> <li>➤ “Tematización Rápida Día Anterior” ver sección “Tematización Rápida Día Anterior”.</li> <li>➤ “Tematización Rápida Semana Actual” ver sección “Tematización Rápida Semana Actual”.</li> <li>➤ “Tematización Rápida Mes Actual” ver sección “Tematización Rápida Mes Actual”.</li> </ul>
<b>Sección “Tematización Libre”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El usuario selecciona el tipo de tematización (rango o fecha exacta), una o varias áreas a tematizar y un criterio de análisis.</p>	<p>2. El sistema muestra dos campos de texto y un botón debajo del campo “Criterio de análisis”. En estos campos el usuario debe especificar un rango de valores deseado, el cual delimitará los datos de consumo que posean las áreas anteriormente seleccionadas (ver Prototipo de Interfaz 5).</p>
<p>3. El usuario llena los campos de texto con los valores del rango deseado y presiona el botón izquierdo del mouse sobre el botón “Generar”</p>	<p>4. El sistema construye cuatro rangos a partir del rango de valores entrados por el usuario y genera un estilo para cada uno. El sistema habilita el campo “Clases” y posibilita al usuario cambiar el número de rangos generados por el sistema (ver Prototipo de Interfaz 5).</p>
<p>5. El usuario selecciona la fecha (o el rango de fechas) en correspondencia con el tipo de tematización seleccionada anteriormente y pulsa el botón “Tematizar”</p>	<p>6. El sistema muestra la tematización según la selección del usuario. (ver Prototipo de Interfaz 6)</p>
<p><b>Sección “Tematización Rápida Día Anterior”</b></p>	
<p><b>Flujo Normal de Eventos</b></p>	

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción "Tematización Rápida Día Anterior".	2. El sistema inicializa los valores que serán enviados al servidor con la fecha del día anterior, y el tipo de tematización "superavit".
	3. El sistema procesa la información y muestra en el mapa la tematización de todas las áreas según el superavit del día anterior (ver Prototipo de Interfaz 6).
<b>Sección "Tematización Rápida Semana Actual"</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción "Tematización Rápida Semana Actual".	2. El sistema inicializa los valores que serán enviados al servidor con un rango de fechas entre el día actual y una semana atrás, y el tipo de tematización "superavit".
	3. El sistema procesa la información y muestra en el mapa la tematización de todas las áreas según el acumulado de superavit de los últimos siete días (ver



	Prototipo de Interfaz 6).
<b>Sección “Tematización Rápida Mes Actual”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción “Tematización Rápida Mes Actual”.	2. El sistema inicializa los valores que serán enviados al servidor con un rango de fechas entre el día actual y un mes antes, y el tipo de tematización “superavit”.
	3. El sistema procesa la información y muestra en el mapa la tematización de todas las áreas según el acumulado de superavit de los últimos treinta días (ver Prototipo de Interfaz 6).
<b>Prototipo de Interfaz</b>	

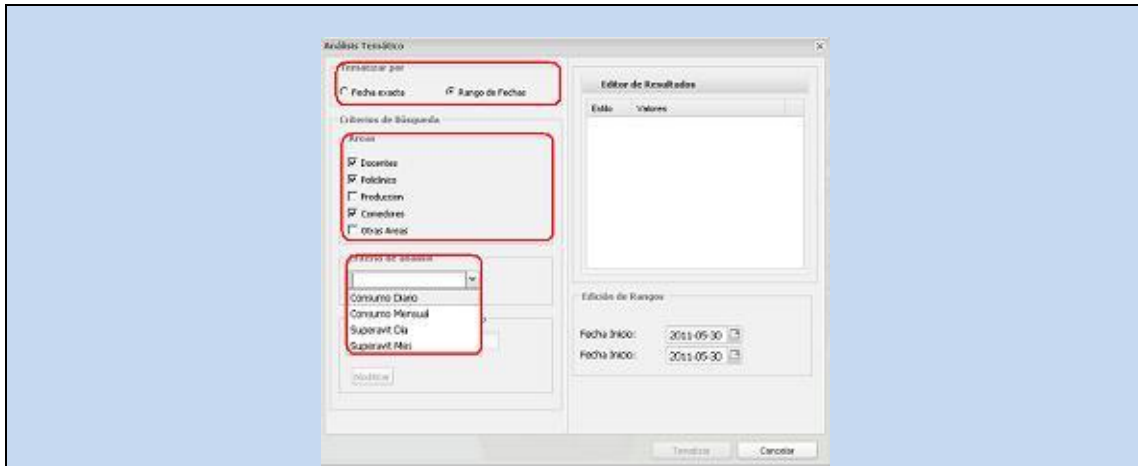


Figura 3. 7.- Prototipo de Interfaz 4

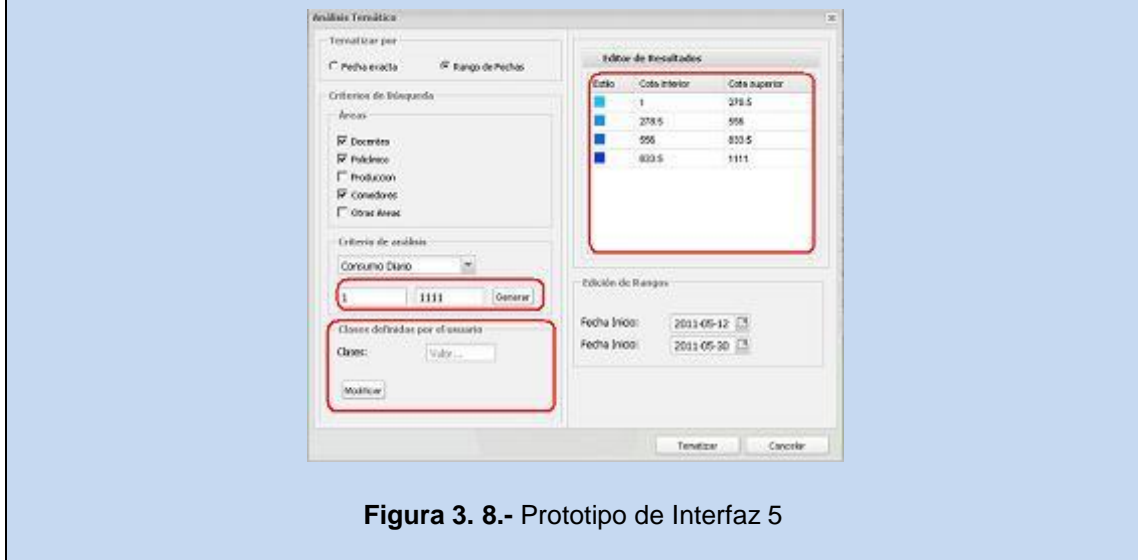


Figura 3. 8.- Prototipo de Interfaz 5



**Figura 3. 9.-** Prototipo de Interfaz 6

**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
<p><b>5.1.</b> El usuario introduce datos inválidos (por ejemplo: el valor máximo menor que el valor mínimo) y presiona el botón “Generar”.</p>	<p><b>6.1.</b> El sistema muestra una alerta indicando que los datos son inválidos. Acto seguido borra los datos seleccionados en el campo “Criterio de análisis”, oculta los campos de texto para introducir el rango y el botón “Generar”, y permite que el usuario seleccione nuevamente estos valores (ver Prototipo de Interfaz 7).</p>
<p><b>7.1.</b> El usuario selecciona una fecha o un rango de fechas inválido.</p>	<p><b>6.1.</b> El sistema muestra una alerta indicando que hubo un error en la entrada de datos (ver Prototipo de Interfaz 7).</p>

## Prototipo de Interfaz

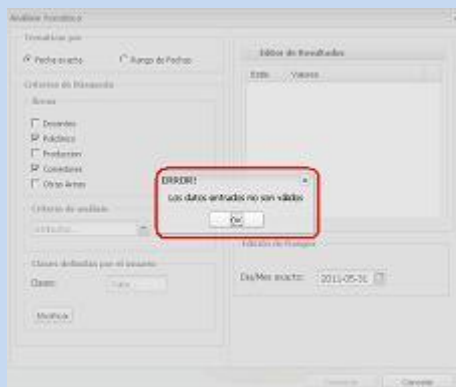


Figura 3. 10.- Prototipo de Interfaz 7

## Poscondiciones

No tiene

### 3.6 Conclusiones parciales

Al finalizar el presente capítulo, se obtiene una mejor documentación acerca de la estructura del sistema a implementar. Se obtuvieron los requisitos funcionales y no funcionales que servirán de guía para la verificación de las funcionalidades del sistema. Además fueron confeccionadas las descripciones de los casos de uso del sistema para guiar el proceso de desarrollo de las principales funcionalidades que tendrá el mismo. Todo lo anteriormente descrito contribuye al proceso de implementación del sistema.

# CAPÍTULO 4. Diseño e implementación de la solución

---

## 4.1 Introducción

A continuación se tratarán los temas referentes a la construcción de la solución. Se ofrece una descripción breve del sistema a través de los principales elementos del diseño e implementación. En la medida en que se avance en la lectura del presente capítulo se podrán encontrar elementos que coadyuvarán a una mejor comprensión del sistema. Entre otros, se encontrarán los diagramas de clases del diseño de los casos de uso del sistema, principales patrones de diseño usados, diagrama entidad relación y de clases persistentes del modelo de datos. Asimismo, se ofrece una breve descripción de los modelos de despliegue y pruebas.

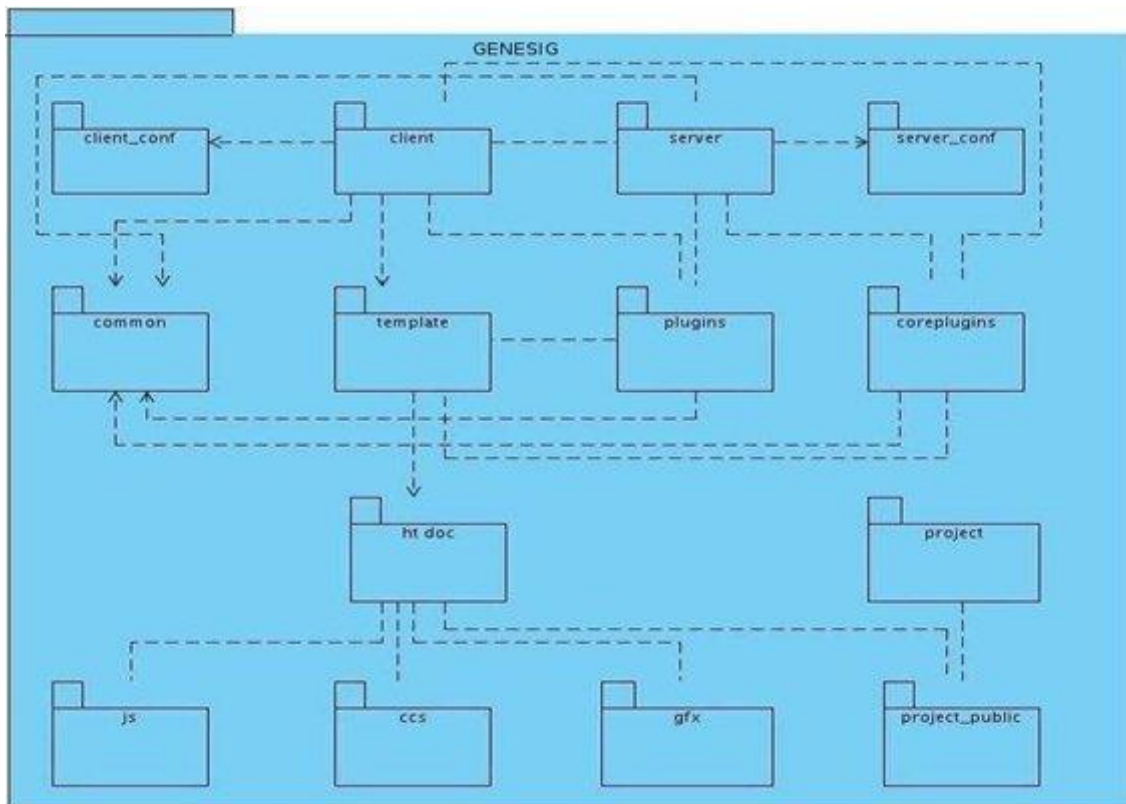
## 4.2 Modelo de diseño

Durante el diseño de la aplicación es necesario identificar los principales elementos de diseño que darán soporte a una alineación más precisa con la arquitectura y las tecnologías escogidas. El modelo de diseño puede ser más o menos cercano al modelo de implementación, dependiendo de la forma de relacionar las clases, paquetes y subsistemas a las clases de implementación, archivos, paquetes y subsistemas del modelo de implementación.

Debido a que la solución se desarrollará a partir de la plataforma soberana GeneSIG- la cual está desarrollada con el framework CartoWeb- es preciso dar una mirada a la estructura lógica del sistema partiendo de la descripción de la arquitectura del mismo y sus principales elementos estructurales. Para ello es necesario ofrecer una breve descripción de la arquitectura de la plataforma.

Al ser desarrollado como un proyecto publicado de CartoWeb, la plataforma mantiene el diseño del framework. Este se encuentra conformado por paquetes con funciones específicas (Figura 4.1), y cuya relación interactiva contribuye a su buen funcionamiento. Entre ellos se encuentra el paquete “*project*”, el cual almacena las aplicaciones que se publican, y es precisamente donde se encuentra publicado el proyecto GeneSIG.

La solución respetará de manera estricta el diseño de la plataforma, pero solo se conservarán los paquetes que sean necesarios para dar solución a las funcionalidades principales. Dichas funcionalidades serán implementadas a través de la creación de componentes – o plugins – para el proyecto publicado GeneSIG.



**Figura 4. 1.-** Elementos arquitectónicamente significativos de GeneSIG

<b>Módulos</b>	<b>Descripción</b>
client	Contiene las clases clientes que utiliza CartoWeb.
server	Contiene las clases servidoras que utiliza CartoWeb.
coreplugins	Contiene los plugins básicos del CartoWeb.
projects	Contiene los proyectos publicados como el GeneSIG.
common	Contiene las clases que son comunes para el funcionamiento del CartoWeb. Incluye las clases que servirán de puente para la comunicación entre cliente-servidor.
plugins	Contiene los plugins de CartoWeb que pueden ser modificados.
templates	Contiene las plantillas que utiliza CartoWeb, construidas en el generador de plantillas Smarty para el procesamiento de datos.
server_conf	Contiene las configuraciones de las clases servidoras.
client_conf	Contiene las configuraciones de las clases clientes.
htdocs	Contiene los ficheros htdocs que utiliza el CartoWeb.
js	Contiene los ficheros js de programación de la interfaz que utiliza CartoWeb.
css	Contiene los estilos que utiliza CartoWeb.

gfx	Contiene las imágenes que utiliza CartoWeb.
project_public	Contiene los vínculos de los proyectos publicados como GeneSIG.

#### 4.2.1 Estructura de un plugin o componente

El sistema está diseñado con una arquitectura basada en componentes, la cual le proporciona una flexibilidad y escalabilidad muy elevadas. Cada uno de los componentes tiene una estructura estándar (Figura 4.2), similar en funcionamiento a la de CartoWeb: un paquete que contiene las plantillas, los paquetes cliente y servidor que se comunican a través de un paquete común; además de un paquete de documentos donde se encuentran la implementación de la interfaz del plugin con documentos Javascript, las imágenes que usará éste y las hojas de estilos.

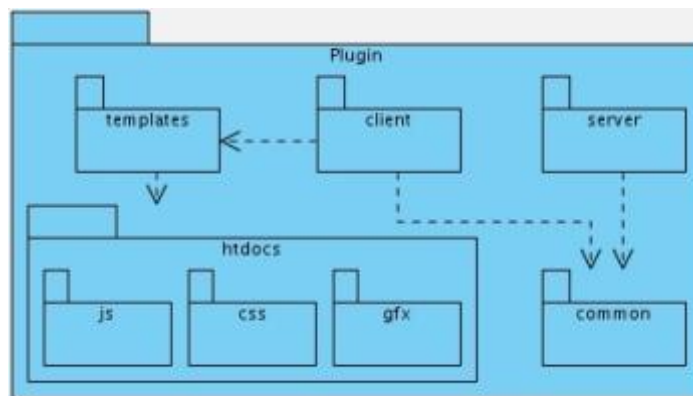


Figura 4. 2.- Estructura de un plugin de CartoWeb

#### 4.2.2 Patrones de diseño

Los patrones de diseño representan soluciones probadas a problemas comunes en la programación. Cada patrón se compone de cuatro elementos principales: el nombre, el problema, la solución y las consecuencias. El nombre sirve para identificarlos unos de otros y ofrecer una idea de su funcionamiento.



El problema consiste en una explicación del contexto en el que se puede aplicar el patrón. La solución es una relación de los elementos del diseño que lo componen, con sus respectivas responsabilidades, relaciones y colaboraciones; esta no contiene una implementación exacta, sino una descripción abstracta de un problema de diseño y cómo una estructura de elementos (clases y objetos) pueden solucionarlo. Por último, las consecuencias representan los resultados de la aplicación del patrón y son el punto crítico a la hora de evaluar alternativas de diseño, así como para la comprensión de la correspondencia costo-beneficio a partir de la aplicación del mismo.

Entre los patrones de diseño más utilizados existen dos grandes grupos, los patrones GOF y los GRASP. Los que han sido incluidos en el diseño de la aplicación pertenecen al primer grupo.

#### **4.2.2.1 Singleton**

Este patrón se agrupa con los demás patrones creacionales, aunque es hasta cierto punto un patrón que limita- en vez de promover- la creación de clases. Específicamente su función es asegurarse de que existe una, y solo una, instancia de una clase; para lograr esto provee un punto de acceso global a la misma. En la solución este patrón es usado para mantener una sola instancia del mapa mientras las clases del servidor y del cliente acceden a ella y modifican a través del paquete “*common*”.

#### **4.2.2.2 Command**

El patrón Command realiza peticiones a objetos específicos. Esta encierra una acción específica dentro de un objeto y lo dota con una interfaz pública. Permite darle al cliente la habilidad de realizar peticiones sin conocer la acción que será realizada; además permite cambiar dicha acción sin afectar al cliente en forma alguna. En el sistema, el usuario está en constante interacción con la interfaz gráfica, para solicitar cualquier información – ya sea de consumo o socioeconómica –, y mediante el uso de este patrón se controla el proceso de comunicación del usuario con el sistema de manera sencilla y centralizada.

#### **4.2.3 Diagrama de clases del diseño**

Para la realización de los diagramas de clases del diseño fueron tomados en consideración solamente los elementos más significativos de CartoWeb y GeneSIG. En estos se presenta la estructura de las principales funcionalidades del sistema y la interacción de sus componentes.

### 4.2.3.1 Diagrama de clases del diseño del CUS Graficar

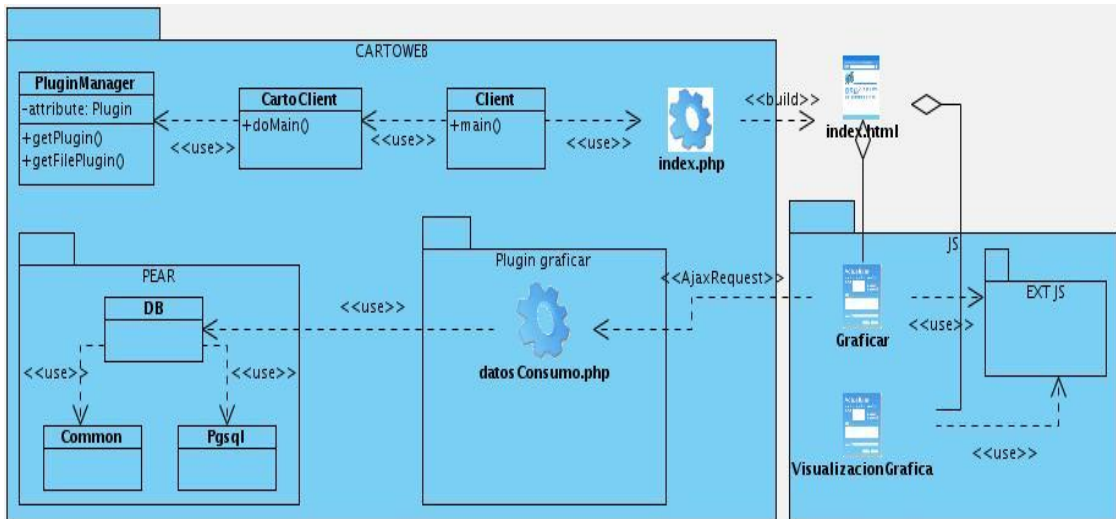


Figura 4. 3.- Diagrama de clases del diseño del CUS Graficar

### 4.2.3.2 Diagrama de clases del diseño del CUS Tematizar

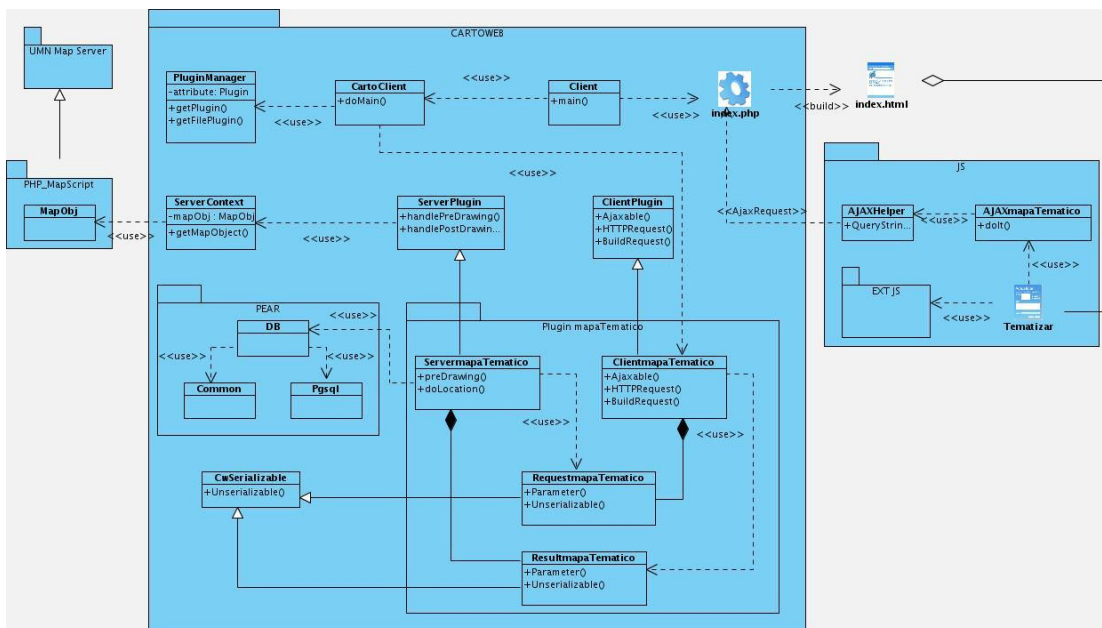


Figura 4. 4.- Diagrama de clases del diseño del CUS Tematizar

### 4.3 Modelo de datos

Se utiliza para diseñar la estructura de los almacenes de datos persistentes que se utilizan en el sistema. El lenguaje UML proporciona un conjunto de elementos de modelado que se pueden utilizar para desarrollar el diseño detallado de las tablas en la base de datos y modelar el diseño del almacenamiento físico de la base de datos. Sirve además para brindar una mejor comprensión de la capa de datos de la aplicación. Incluye los esquemas de bases de datos, y las estrategias de transformación entre representaciones de objetos y no objetuales.

Este proceso debe ser desarrollado de manera correcta pues un modelo de datos erróneo podría introducir riesgos en el desarrollo del sistema. Esto se debe a que la estabilidad de la arquitectura del sistema depende esencialmente del modelado de datos y el desarrollo de un prototipo arquitectónico que permita que el rendimiento de la base de datos se evalúe.

#### 4.3.1 Diagrama Entidad-Relación

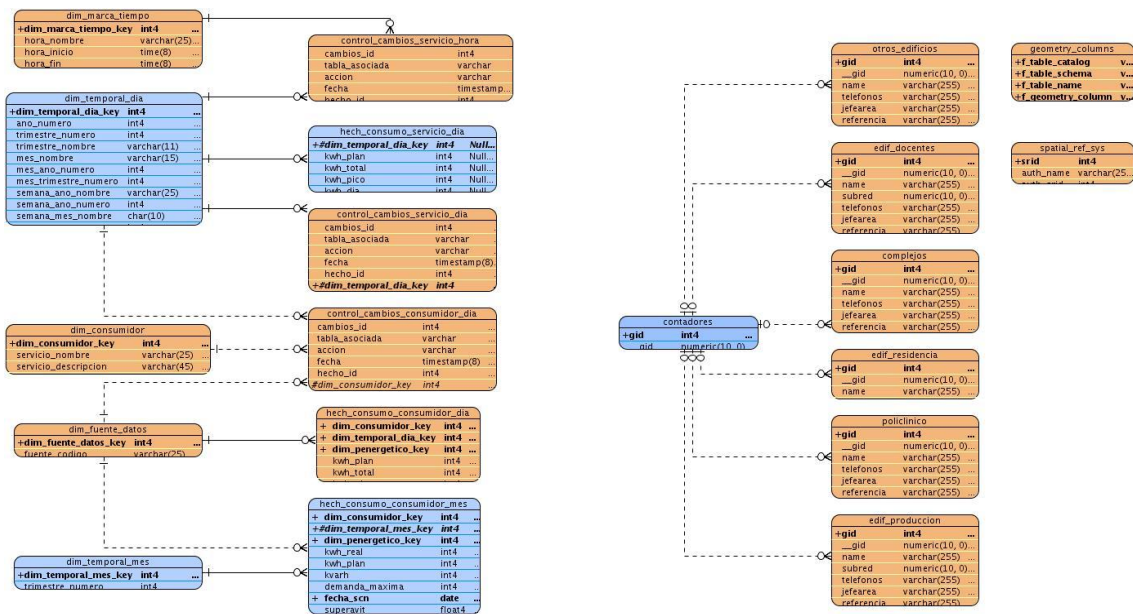


Figura 4. 5.- Diagrama Entidad-Relación



implementación física y el diseño lógico, comúnmente usado en un enfoque de ingeniería directa e inversa donde se sincronizan los archivos de código fuente con un modelo de diseño/implementación combinado.

Los artefactos UML creados durante el trabajo de diseño -los diagramas de clases del diseño- se utilizarán como entradas en el proceso de generación de código. Éste contiene los artefactos de implementación como el código fuente, las definiciones de bases de datos, las páginas, etcétera. Por tanto, el código que se va a crear forma parte del Modelo de Implementación.

#### 4.4.1 Modelo de despliegue

Contiene el despliegue físico de los procesos y componentes sobre los nodos, así como la configuración de la red física entre ellos. Ofrece una vista sobre el modelo de despliegue del proceso unificado, visualizada con los diagramas de despliegue de UML. Normalmente, la "vista" es simplemente el modelo completo en lugar de un subconjunto, ya que todo es relevante.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. Es un conjunto de nodos unidos por conexiones de comunicación donde un nodo puede contener instancias de componentes software, objetos y procesos. El diagrama de despliegue de la solución muestra dichas conexiones y los componentes necesarios para el correcto funcionamiento de la misma. En éste se pueden encontrar los clientes como dispositivos (PC, Laptop), los servidores (web y de mapas) – los cuales serán accedidos a través del protocolo HTTP – y las bases de datos del sistema- a las cuales estarán disponibles a través del protocolo TCP/IP.

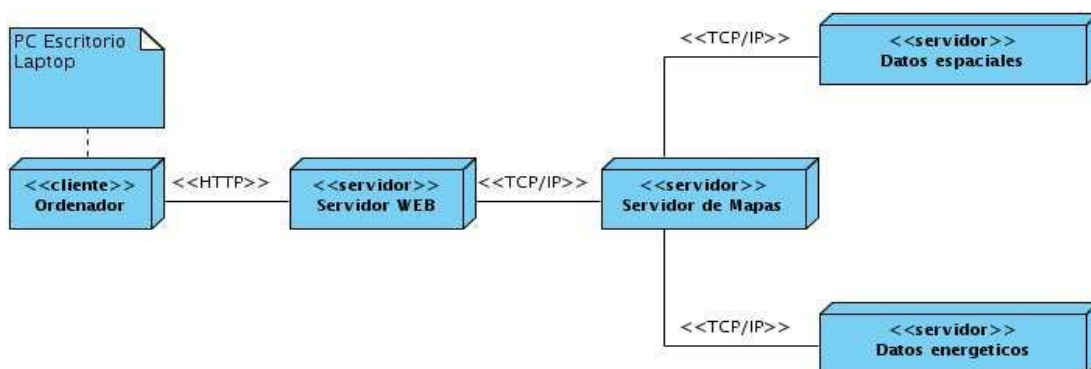


Figura 4. 7.- Diagrama de despliegue de la solución

## **4.5 Plan de pruebas**

Para el usuario nada es más importante que el software cumpla con sus expectativas. De esta forma, la satisfacción del usuario requiere – en cierto momento – de la verificación de los requisitos mediante los casos de prueba. Debido a que dicha verificación puede ser realizada de varios modos y por distintos verificadores, y a la imposibilidad de verificar todos los requisitos, resulta de extrema importancia para el éxito del proyecto la selección de los requisitos más adecuados para la prueba. Estos deberán representar un equilibrio entre el coste, el riesgo y la necesidad de tener el requisito verificado.

Debido a que se tiene conocimiento de las funcionalidades específicas del software, pueden ser realizadas pruebas que demuestren la operatividad de las funcionalidades. Para llevar a cabo este tipo de pruebas será utilizado el enfoque de caja negra, basado en la interacción con la interfaz de la aplicación. Para corroborar el correcto funcionamiento del sistema, los casos de prueba consistirán principalmente en validar que la entrada se acepta de forma adecuada y que se produce la salida esperada.

El proceso de pruebas de caja negra estará dirigido principalmente a la confirmación de los requisitos funcionales del software y a comprobar tanto el comportamiento de la interfaz gráfica, como su interacción con el usuario y la calidad funcional del sistema. A diferencia de las pruebas de caja blanca, éstas se realizan sin tener en cuenta la estructura interna del software.

### **4.5.1 Objetivos**

El objetivo de realizar este tipo de pruebas al sistema es detectar el incorrecto o incompleto funcionamiento del mismo, así como los errores de interfaces, de rendimiento, errores de inicialización y terminación.

### **4.5.2 Descripción**

Como parte del enfoque de caja negra, la técnica de particiones equivalentes es una de las más efectivas. Esta técnica permite examinar los valores válidos e inválidos de las entradas y permite descubrir errores que, de otro modo, requerirían la ejecución de una serie de casos antes de detectar un error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores,

reduciendo así el número de clases de prueba necesarias. El usado para diseñar los casos de pruebas es una variante de la técnica antes descrita.

#### **4.5.3 Resultados de las pruebas**

Una vez realizadas las pruebas, es posible afirmar que el sistema posee todas las funcionalidades propuestas. Además, cada una de ellas fue correctamente validada y – en general– cumplen con los requerimientos especificados por el cliente.

#### **4.6 Conclusiones parciales**

Con este capítulo se obtiene completamente diseñada y construida la solución. Asimismo se obtiene una vista lógica del sistema a partir de sus componentes y subsistemas de implementación, lo cual provee una noción más profunda de su estructura interna y funcionamiento. Además se diseñó e implementó el plan de pruebas a las funcionalidades críticas, y se obtuvieron resultados positivos. A partir del análisis de las funcionalidades del sistema, fue posible llegar a la conclusión de que la aplicación cumple con los requerimientos funcionales y no funcionales obtenidos en la captura de requisitos, y está lista para comenzar el proceso de liberación.

## CONCLUSIONES

---

Una vez concluida la presente investigación se consideran cumplidos los objetivos que dieron inicio al desarrollo la misma. El uso de los métodos teóricos seleccionados contribuyó a una mejor comprensión del objeto de estudio y el contexto del problema. Además, la correcta definición de las tareas y objetivos investigativos contribuyó a una mejor organización de la investigación. Se obtuvieron todos los artefactos que documentan el proceso desarrollo y la documentación técnica asociada al sistema

Al finalizar el proceso de desarrollo se cuenta con un Sistema de Información Geográfica capaz de georreferenciar las áreas de la universidad y manejar la información de las que posean contadores eléctricos asociados a ellas. Este sistema cuenta con funcionalidades que introducen una mejora en el proceso de supervisión del consumo energético en la Universidad de las Ciencias Informáticas, debido a la posibilidad que brinda a la Dirección Energética de consultar los datos de consumo asociados a los grandes consumidores de la universidad, así como provee un mayor volumen de información. Esto se traduce en una mejor planificación y una potencialidad en la toma de medidas de carácter urgente. El sistema cumple con los requerimientos especificados para su desarrollo, por lo cual se concluye que cumple con los resultados esperados.

Como ha sido descrito en capítulos anteriores, la UCI –debido a las tecnologías existentes y al alto grado de informatización– presenta niveles de consumo que sobrepasan con creces los de las restantes universidades y algunas empresas. El desarrollo del sistema SIG Energía, por tanto, constituye un aporte importante para la universidad, pues permite la optimización del proceso de supervisión del consumo para una mejor planificación y toma de medidas, lo cual constituye una herramienta más para dar cumplimiento a las políticas de ahorro de la entidad y del país.



## RECOMENDACIONES

---

Teniendo en cuenta los beneficios que proporciona el presente trabajo investigativo y las experiencias acumuladas durante el proceso de desarrollo del software y el transcurso de la investigación, se recomienda:

- Desplegar el sistema en la Universidad de las Ciencias Informáticas para su uso por los especialistas de la Dirección Energética de la misma.
- Implementar una nueva versión con mayor cantidad de funcionalidades que incluya una actualización automática de los datos.
- Efectuar un análisis de factibilidad sobre el desarrollo de una aplicación de escritorio con funcionalidades similares a la solución, y – de ser factible – realizar la implementación de un Sistema de Información Geográfica de escritorio que incluya características que le permitan realizar análisis de los congestionamientos de la red, determinar áreas de crecimiento de la misma, mostrar registros de consumo por hora del día, entre otros.

## TRABAJOS CITADOS

---

- Amador Guerra, J. (2000). *Análisis de los parámetros técnicos en la aplicación de los sistemas de información geográfica a la integración regional de las energías renovables en la producción descentralizada de electricidad*.
- Autores, C. D. (2009). *Elementos de UML*. Recuperado el 6 de Diciembre de 2010, de KDE Docs: <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>
- Bosque, J. (1992). *Sistemas de Información Geográfica*. Madrid: RIALP.
- CIS DGroup. (2007). *Centralized Integrated System*. Recuperado el 17 de Enero de 2011, de Centralized Integrated System: <http://www.cis-dgroup.com/Index.html>
- Domínguez, J. (1994). Conceptos básicos y aplicaciones de los Sistemas de Información Geográfica. *Technical Desktop Magazine* , 28-33.
- Easton, C. (2001). *Advanced Javascript*. Wordware Publishing, Inc.
- Echevarría, M. (2004). *Arquitectura de software*. Recuperado el 27 de Febrero de 2011, de Scribd: <http://www.scribd.com/doc/31552918/Arquitectura-clase3>
- ETSI Topografía, Geodesia y Cartografía. (2008). *Departamento de Ingeniería Topográfica y Cartografía UPM*. Recuperado el 2 de Febrero de 2011, de Departamento de Ingeniería Topográfica y Cartografía UPM: <http://www.geo.upm.es/ven/resources/glossary.html>
- Franco, R. (1 de Julio de 2007). *Sistemas de Información Geográfica*. Recuperado el 17 de Enero de 2011, de Sistemas de Información Geográfica: <http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/intro.htm#HISTORIA>
- Guinea de Salas, A. (18 de Junio de 2007). *¿Qué es la Geoinformática?* Recuperado el 21 de Enero de 2011, de Directorio electrónico de Guatemala: [http://www.deguate.com/artman/publish/gestion\\_articulos/Qu\\_es\\_la\\_Geoinform\\_tica\\_9565.shtml](http://www.deguate.com/artman/publish/gestion_articulos/Qu_es_la_Geoinform_tica_9565.shtml)
- Hernandez, Y., & Sanchez, H. (2009). *Gestión de Nomencladores para el Programa Nacional de Informatización del Conocimiento Geológico*. La Habana.
- Jacobson, I. (2000). *El Proceso Unificado de Desarrollo de Software*. Adison Wesley Series.
- La Torre, L., Argota, I., & Fung, J. (2009). *SCADA Guardián del ALBA*. La Habana: Informática 2010.

Mitchell, T. (2005). *Web Mapping Illustrated*. O'Reilly.

Puertas, J. (2002). *Un paseo por PHP*. Almería: Universidad de Almería.

Salinas, J. (2007). *Información Geográfica, Software Libre e Infraestructuras de Datos Espaciales*. La Habana.

## BIBLIOGRAFÍA

---

Amador Guerra, J. (2000). *Análisis de los parámetros técnicos en la aplicación de los sistemas de información geográfica a la integración regional de las energías renovables en la producción descentralizada de electricidad*.

Autores, C. D. (2009). *Elementos de UML*. Recuperado el 6 de Diciembre de 2010, de KDE Docs: <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>

Borges, D. (1996). *A geographical information system for large scale integration of renewable energies into regional energy markets*. Florencia: EDIFIR.

Bosque, J. (1992). *Sistemas de Información Geográfica*. Madrid: RIALP.

Cebrián, J. (1992). *Información geográfica y sistemas de información geográfica*. Santander: DIDOTSA.

CIS DGroup. (2007). *Centralized Integrated System*. Recuperado el 17 de Enero de 2011, de Centralized Integrated System: <http://www.cis-dgroup.com/Index.html>

Comas, D., & Ruiz, E. (1993). *Fundamentos de los sistemas de información geográfica*. Barcelona: Ariel.

Domínguez, J. (1994). Conceptos básicos y aplicaciones de los Sistemas de Información Geográfica. *Technical Desktop Magazine*, 28-33.

Easton, C. (2001). *Advanced Javascript*. Wordware Publishing, Inc.

Echevarría, M. (2004). *Arquitectura de software*. Recuperado el 27 de Febrero de 2011, de Scribd: <http://www.scribd.com/doc/31552918/Arquitectura-clase3>

ETSI Topografía, Geodesia y Cartografía. (2008). *Departamento de Ingeniería Topográfica y Cartografía UPM*. Recuperado el 2 de Febrero de 2011, de Departamento de Ingeniería Topográfica y Cartografía UPM: <http://www.geo.upm.es/ven/resources/glossary.html>

Federane, J. (1996). A geographical information system for large scale integration of renewable energies into regional energy markets. *Second Interim Report-Enero*. Bruselas: Comisión de las Comunidades Europeas.

Franco, R. (1 de Julio de 2007). *Sistemas de Información Geográfica*. Recuperado el 17 de Enero de 2011, de Sistemas de Información Geográfica: <http://gemini.udistrital.edu.co/comunidad/profesores/rfranco/intro.htm#HISTORIA>

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1998). *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley Series.
- Guinea de Salas, A. (18 de Junio de 2007). *¿Qué es la Geoinformática?* Recuperado el 21 de Enero de 2011, de Directorio electrónico de Guatemala: [http://www.deguate.com/artman/publish/gestion\\_articulos/Qu\\_es\\_la\\_Geoinform\\_tica\\_9565.shtml](http://www.deguate.com/artman/publish/gestion_articulos/Qu_es_la_Geoinform_tica_9565.shtml)
- Gutiérrez, J., & Gould, M. (1994). *SIG: Sistemas de información geográfica*. Madrid: Síntesis.
- Hernandez, Y., & Sanchez, H. (2009). *Gestión de Nomencladores para el Programa Nacional de Informatización del Conocimiento Geológico*. La Habana.
- Jacobson, I. (2000). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley Series.
- La Torre, L., Argota, I., & Fung, J. (2009). *SCADA Guardián del ALBA*. La Habana: Informática 2010.
- Membrides, A. (2008). *Como desarrollar plugins para CartoWeb*.
- Mitchell, T. (2005). *Web Mapping Illustrated*. O'Reilly.
- Muselli. (1997). A geographical information system for the integration of stand-alone PV systems in remote areas. *14th European Photovoltaic Solar Energy Conference. Volumen II*. Barcelona: H. S. Stephens & Associates.
- Pressman, R. (2005). *Ingeniería del Software. Un Enfoque Practico*. Addison Wesley Series.
- Puertas, J. (2002). *Un paseo por PHP*. Almería: Universidad de Almería.
- RAE. (2010). *Diccionario de la Real Academia Española*. Recuperado el 31 de Enero de 2011, de Real Academia Española: <http://buscon.rae.es/drae/>
- Salinas, J. (2007). *Información Geográfica, Software Libre e Infraestructuras de Datos Espaciales*. La Habana.
- Serra del Pozo, P. (Junio de 2008). *Revista Internacional de Ciencias de la Tierra*. Recuperado el 6 de Febrero de 2011, de Revista Internacional de Ciencias de la Tierra: [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=179](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=179)
- Silva, D. J. (2005). *Aplicación de sistemas de información geográfica en Cuba*. Recuperado el 15 de Enero de 2011, de Mapping Interactivo: [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1051](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051)
- Star, J., & Estes, J. (1990). *Geographic Information Systems: An Introduction*. Prentice Hall.