UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD DE LAS CIENCIAS Y TECNOLOGÍAS COMPUTACIONALES

Título: Sistema de Información Geográfica para la gestión de información de los parques eólicos en Cuba. (SIG_GIPEC)

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

Autores:

Maibol Stable Bientz

Chabelys Marlen Lestapier Lao

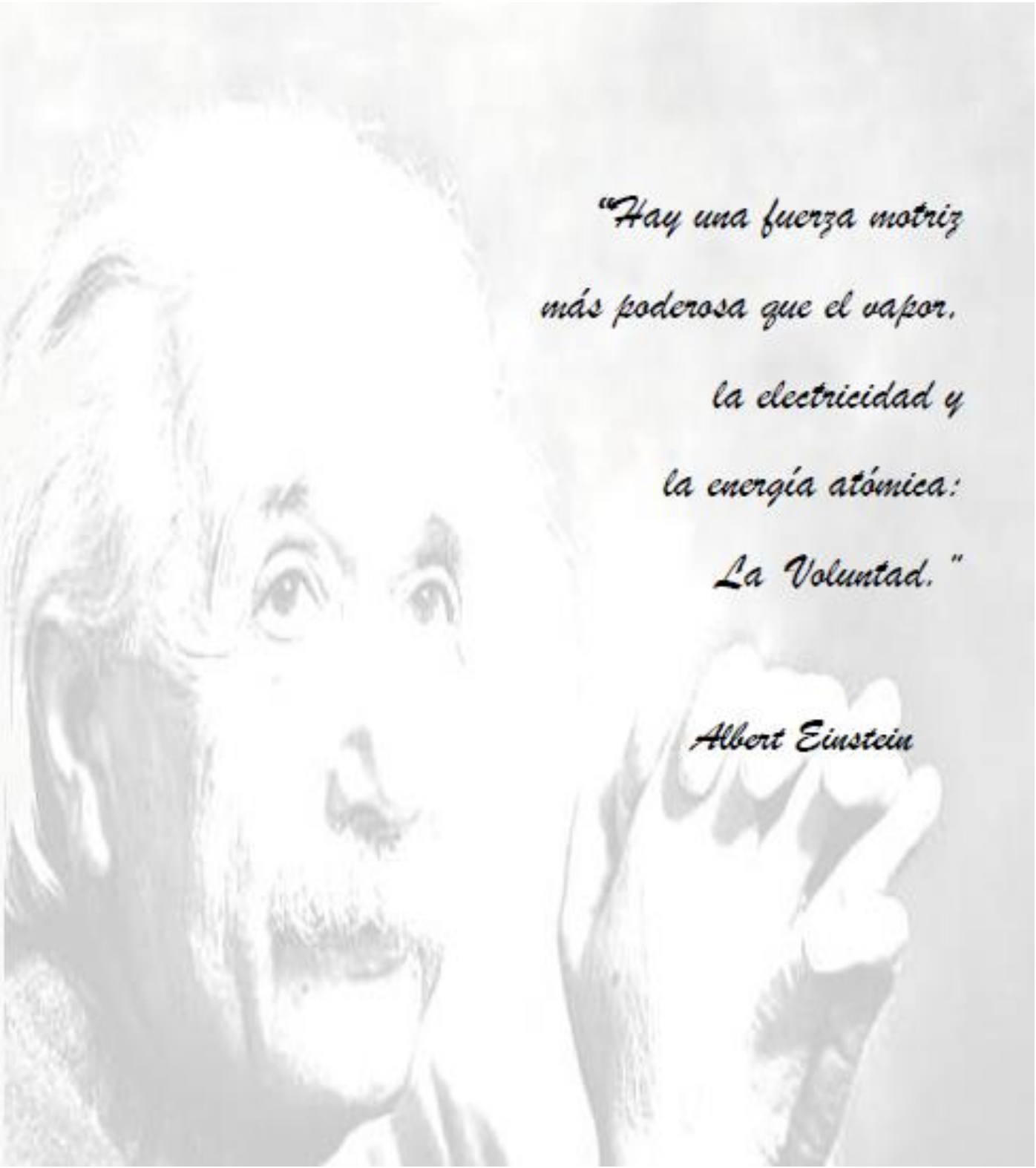
Tutor(es): MSc. Daniel Echevarría González

Co-Tutor(es): Ing. Camilo Madariaga Dozaret

Ing. Isleivys Rodríguez Pupo

La Habana, mayo de 2017

¨Año 59 de la Revolución¨


"Hay una fuerza motriz
más poderosa que el vapor,
la electricidad y
la energía atómica:
La Voluntad."
Albert Einstein

Declaración de autoría

Declaramos ser únicas autoras del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autores: _____ _____

Maibol Stable Bientz Chabelys M. Lestapier Lao

Tutor: _____

MSc. Daniel Echevarría González

Datos de contacto

Autores: Maibol Stable Bientz

Chabelys M. Lestapier Lao

Correo Electrónico: mstable@estudiantes.uci.cu

cmlestapier@estudiantes.uci.cu

Tutor: MSc. Daniel Echevarría González

Formación Académica: Graduado de la Facultad de Geografía de la Universidad de La Habana, Master en Ciencias en la Facultad de Geografía de la U/H en el año 1998. Profesor Asistente en el año 2012.

Centro Laboral: LPS Aplicativos SIG, Universidad de las Ciencias Informáticas (UCI).
Correo Electrónico: danielec@uci.cu

Co_Tutora: Ing. Isleivys Rodríguez Pupo

Formación Académica: Graduada en la Universidad de las Ciencias Informáticas. Ingeniera en Ciencias Informáticas en el año 2013.

Cargo: Asesora de Planificación y Control.

Centro Laboral: Centro de Geoinformática y Señales Digitales, Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: irodriguez@uci.cu

Co_Tutor: Ing. Camilo Madariaga Dozaret

Formación Académica: Graduado en la Universidad de las Ciencias Informáticas en el año 2014.

Centro Laboral: LPS Aplicativos SIG, Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: cdozaret@uci.cu

Dedicatoria

A mis padres y hermano por vivir junto a mí,

este viaje que ha sido la UCI.

A mi familia por el apoyo y el amor.

A mis amigos, que no son pocos.

A mis abuelos, Francisco y Pedro, que me cuidan siempre aunque ya no están.

De Maibol

A mis abuelos Juana y Bombino, que aunque ya no estén conmigo siempre han sido mis guías.

A mi mama por su amor incondicional y sacrificio.

A mi primo Tommy por ponerme el listón bien alto y

a mi hermano Bryant por ser quien me inspiró a coger Informática.

A mi amor por estar siempre a mi lado y por su paciencia.

De Chabelys

Agradecimientos

A mi más entrañable amiga, mi madre, por mostrarme siempre el camino, no desistir ante mis equivocaciones y amarme como solo una madre puede.

A mi padre, por estar siempre ahí cuando lo necesito y cuando no, guiarme, aconsejarme y amarme tanto, aunque no lo expreses, pues sé que lo haces.

A mi hermano por ser mi segundo padre y mi amigo, te amo.

A mi familia, que se están orgullosos de mí por haber llegado tan lejos, ustedes tuvieron mucho que ver.

A mis amigos, dentro y fuera de la UCI, que lloran y ríen conmigo, me alegro de que estén en mi vida, en especial Tania, Viunaiki, "Mi gorda" y "Meñique" que no han permitido que el tiempo y lo agitado de la vida nos separe, son mi segunda familia.

A todas las personas que han trabajado tan arduamente para que este proyecto se materializara, en especial mis tutores, pues sé que dimos trabajo, gracias.

A mi compañera de tesis, por soportarme, tener paciencia y sobre todo cariño y amor de sobra, gracias por ser tan tú.

Maibol

Agradecimientos

Agradezco a mi mamuchi, Marlen, por su amor, sacrificio y apoyo incondicional. Por ser mi mejor amiga, por inculcarme valores y por hacer de mí la mujer que soy hoy. A mis abuelos Juana Núñez y Cándido Bombino, por ser mis pilares, por llevarme bien tempranito a la escuela y por darme todo el amor que me dieron. Los extraño.

A mi papa Guillermo por todo su apoyo y estar siempre pendiente de mí, al igual que mis abuelos Yolanda y Papin.

A mis papas Julian y Pablo, por estar ahí siempre que los necesito y aportar sus cubos de arena en lo que soy hoy.

A primo Tommy por ponerme el listón bien alto, por ser mi ejemplo y por su ayuda, que sin el hoy no estaría aquí.

Al tatico, mi hermano Bryant, por ser la inspiración, por sus locuras y por quererme mucho. A mi familia en general.

Al amor de mi vida, Ariel, por estar siempre a mi lado, por su paciencia y amor. Por regalarme tres años súper especiales y sigo contando. Por ser lo mejor que me ha pasado en la vida. Te amo.

Agradecer también todas las amistades que la UCI me doy la oportunidad de conocer, sobre todo a dos de mis mejores amigas Deby y Eli. Por todos los momentos que pasamos, por ser mi apoyo moral y mis paños de lágrimas.

A Julio, a Yanara y Claudia por ser parte también de ese club de amistades que quiero un montón. A mi compañera de tesis, Maibol, que me ha soportado todo este año sin quejarse mucho, gracias por tu amistad y tu confianza. A Viunaiki y a Tania, por aceptarme y quererme. Por los momentos de risas, por la paciencia y por todo lo que hemos compartido. De manera general a todos mis compañeros de clases por compartir estos años de experiencias y amistad. Los llevare siempre conmigo.

Agradecerles a mis tutores Daniel, Camilo e Isleivys por su ayuda y su tiempo. Además especialmente agradecerle a mi primer tutor Ariel Labrada que me ayudo a iniciar la tesis y sin ese impulso no creo que lo hubiese conseguido.

A Adrián Gracia y de nuevo a Eli por ser mis tutores externos, realmente, gracias por su ayuda.

A todos, los que mencione y los que se me quedaron fuera, gracias por todo y por formar parte de mi vida.

Chabelys

RESUMEN

El aprovechamiento de fuentes de energía renovables se ha convertido en una alternativa necesaria en todo el mundo con el propósito de reducir la contaminación ambiental y los costos de la energía fósil. En Cuba, la Empresa de Ingeniería y Proyectos para la Electricidad (INEL), está especializada en el análisis, desarrollo y mantenimiento de parques demostrativos eólicos. En dicha empresa se llevan a cabo estudios, en función de analizar y controlar dichos parques, y áreas potencialmente viables para el establecimiento de nuevos parques. Actualmente se trabaja con datos de estudios previos al año 2010, por lo que no se cuenta con información actualizada, tanto de las zonas prospecto como de la generación real de los parques eólicos. Se hace necesaria la implementación de un sistema que permita tener la información de manera centralizada y accesible para la toma de decisiones. La presente investigación fundamenta el uso de un Sistema de Información Geográfica (SIG) para gestionar la información de la producción energética de los parques eólicos en función de las características nominales de las máquinas y las zonas prospecto. Además, contribuirá en la accesibilidad a los datos y mejorará la toma de decisiones referente a la información sobre los parques eólicos.

Palabras claves: Sistema de Información Geográfica, parques eólicos, zonas prospecto.

ABSTRACT

The use of renewable energy sources has become a necessary alternative around the world for the purpose of reducing environmental pollution and fossil energy costs. In Cuba, the Company of Engineering and Projects for Electricity (INEL), is specialized in the analysis, development and maintenance of wind demonstration parks. In this company, studies are carried out, in order to analyze and control these parks, and potentially viable areas for the establishment of new parks. We are currently working with data from studies prior to 2010, so there is no up-to-date information on both the prospect areas and the actual generation of wind farms. It is necessary to implement a system that allows the information to be centralized and accessible for decision making. The present investigation is based on the use of a Geographic Information System (GIS) to manage the information on the energy production of wind farms according to the nominal characteristics of the machines and the prospect areas. In addition, it will contribute to data accessibility and improve decision-making regarding information on wind farms.

Keywords: Geographic Information System, wind farms, prospect areas.

ÍNDICE

INTRODUCCIÓN 12

CAPÍTULO 1. Fundamentación teórica asociada a la gestión de la información de los parques eólicos. 16

1.1 Introducción. 16

1.2 Sistemas de Información Geográfica. 16

1.3 La gestión de la información de parques eólicos. 17

1.4 Conceptos asociados a la gestión de la información de los campos eólicos. 17

1.5 Importancia de la gestión de la información de la energía eólica en el mundo y Cuba. 19

1.6 Metodología seleccionada para desarrollar la solución. 22

1.7 Herramientas propuestas para dar solución al problema. 23

1.8 Conclusiones del capítulo. 30

CAPÍTULO 2. Diseño e implementación de la solución propuesta. 31

2.1 Introducción 31

2.2 Especificación de requisitos. 31

2.3 Historia de Usuario. 40

2.4 Arquitectura de software. 43

2.5 Modelo de diseño. 44

2.6 Modelo de datos. 47

2.7 Modelo de despliegue. 48

2.8 Diagrama de componentes. 49

2.9 Estándar de codificación 49

2.10 Conclusiones del capítulo. 49

3.1 Introducción. 51

3.2 Pruebas al sistema. 51

3.3 Diseños de Casos de Prueba. 52

3.4 Resultado de las pruebas 53

3.5 Conclusiones parciales. 55

CONCLUSIONES GENERALES 56

RECOMENDACIONES 57

REFERENCIAS BIBLIOGRÁFICAS 58

BIBLIOGRAFÍA 61

ANEXOS 68

Índice de tablas

Tabla 1: Características de los usuarios finales de la aplicación. **37**

Tabla 2: HU_ Realizar identificación puntual **40**

Tabla 3: HU_ Adicionar zona prospecto. **41**

Tabla 4: DCP_Gestionar proveedor: SC1_Adicionar proveedor. **52**

Tabla 5: Conjunto de no conformidades detectadas. **54**

Índice de figuras

Figura 1: Diagrama de clases de diseño: Localizar por elementos.....45

Figura 2: Modelo Entidad-Relación de la base de datos.....48

Figura 3: Modelo de despliegue.....49

Figura 4 : Modelo de diagrama de componentes. 49

Figura 5: Declaración de una clase empleando UpperCamelCase.....50

Figura 6: Declaración de variables empleando lowerCamelCase 50

Figura 7: Resultados de la prueba de caja negra. 54

Figura 8: Inicio de sesión.....68

Figura 9: Sistema de Información Geográfica para la gestión de la información de los parques eólico (SIG_GIPEC). 69

Figura 10: Tematización por colores. 69

Figura 11: Gestionar proveedor.....70

Figura 12: Cálculo de área y perímetro 70

INTRODUCCIÓN

Históricamente el hombre ha empleado distintas fuentes de energías no renovables como el petróleo, el gas natural, el carbón y la energía nuclear; pero los niveles de consumo de las mismas han deteriorado el medio ambiente, reforzando significativamente el cuestionamiento del modelo energético basado en los combustibles fósiles.

Según estudios realizados por el Centro de Análisis de Información del Dióxido de Carbono, de la División de Ciencias Ambientales del Laboratorio Nacional de Oak Ridge, Tennessee EEUU, el aporte de Dióxido de Carbono (CO_2) por la combustión de los recursos no renovables, ha crecido aceleradamente con el paso del tiempo y el constante desarrollo tecnológico y social del mundo.

Ante tales tendencias, se impone la necesidad de una reestructuración energética global que esté basada en el paradigma energético sostenible, utilizando fuentes renovables de energía (FRE) como: solar, eólica, mareomotriz, geotérmica, hidráulica y biomasa. Cabe añadir que la energía eólica ha tenido un auge importante pues constituye la FRE con mayor grado de desarrollo en la actualidad; posicionándola como una forma de energía rentable.

Cuba es productor de petróleo pesado con un alto componente de azufre, generándose un elevado volumen de (CO_2) que afecta significativamente el medio ambiente. Como solución a esta agravante, se han desplegado proyectos orientados a la investigación, evaluación, e implementación de soluciones que satisfagan el objetivo de la Revolución Energética y tributen al Sistema Electroenergético Nacional (SEN) (Gámez, 2012). En busca de dar cumplimiento a dichas acciones se ha tratado de impulsar el trabajo de institutos especializados en esta temática como el Departamento de Fuentes Renovables de la Empresa de Ingeniería y Proyectos para la Electricidad (INEL), enfocado en el análisis, desarrollo y mantenimiento de parques demostrativos eólicos e identificación de áreas potencialmente viables para el establecimiento de nuevos parques.

Actualmente el departamento no cuenta con información actualizada diariamente de la generación real de los parques eólicos, dado que no llega de forma directa y sistemática, pues queda en poder de los operadores de los parques o el Despacho General de Carga. Sobre las zonas de desarrollo, se tienen estudios previos al año 2010 que permitieron hacer una caracterización sintética de las particularidades de las áreas. Además no se cuenta con herramientas informáticas, que permitan gestionar la información de los emplazamientos de los parques y donde se encuentran las zonas prospecto.

Todos los factores anteriormente descritos provocan un retardo en la llegada de la información a INEL, referente a los parques eólicos, dificultándoles: el posible desarrollo de nuevos proyectos, la toma de

decisiones, y la colaboración con otros países interesados en invertir en el impulso de la energía eólica en la isla.

Partiendo de la situación problemática y de la necesidad de mantener una constante actualización de datos tanto de las zonas de desarrollo como de los parques eólicos, se identifica el siguiente **problema a resolver** ¿Cómo facilitar la accesibilidad a la información vinculada a los parques eólicos en Cuba, para apoyar los procesos de toma de decisiones?

Se define como **objeto de estudio** los Sistemas de Información Geográfica para el manejo de información de los parques eólicos enmarcado en el **campo de acción** desarrollo de un Sistema de Información Geográfica para el manejo de información de los parques eólicos en Cuba.

Para dar solución al problema expuesto anteriormente, se plantea como **objetivo general** desarrollar un Sistema de Información Geográfica que facilite la accesibilidad a la información vinculada a los parques eólicos en Cuba para apoyar los procesos de toma de decisiones.

Durante el transcurso de esta investigación se sostienen las siguientes **preguntas científicas**:

- ¿Qué referentes teórico-metodológicos sustentan el proceso de gestión de la información vinculada a los parques eólicos?
- ¿Qué características tiene la creación de un Sistema de Información Geográfica para los parques eólicos en Cuba?
- ¿Cómo organizar el proceso de desarrollo de un Sistema de Información Geográfica para la gestión de la información de los parques eólicos en Cuba?
- ¿El sistema desarrollado favorece el proceso de toma de decisiones de los directivos del departamento de Recursos Renovables perteneciente a la Empresa de Ingeniería y Proyectos para la Electricidad?

Para guiar la investigación y brindar solución al problema planteado anteriormente, se proyectan las siguientes **tareas de investigación**:

1. Definición del marco teórico-metodológico de la investigación.
2. Análisis de sistemas similares existentes en el mundo con el fin de obtener referencias para la confección de la solución propuesta.
3. Caracterización de las herramientas y tecnologías a utilizar en el proceso de desarrollo del Sistema de Información Geográfica, que faciliten la accesibilidad a la información para apoyar los procesos de toma de decisiones.

4. Implementación de las funcionalidades definidas para el sistema.
5. Validación del sistema propuesto para la verificación del cumplimiento de las funcionalidades de dicha aplicación.

Posterior al cumplimiento de las tareas de la investigación se espera tener como posible resultado, un SIG que facilite la accesibilidad a la información para apoyar los procesos de toma de decisión en la gestión de la información de los parques eólicos en Cuba ante determinadas situaciones que lo requieran, así como la documentación ingenieril asociada al desarrollo del sistema.

Con el objetivo de guiar la investigación se tuvieron en cuenta los **métodos científicos** (González, 2012), que proporcionaron la orientación y dirección adecuada al trabajo del investigador y condicionaron los nuevos conocimientos referentes al trabajo que se realizará. A continuación, se describen los métodos utilizados.

Métodos teóricos: su contenido surge de la experiencia con cierta elaboración racional. El reflejo del objeto está limitado a la contemplación sensorial y los datos que se obtienen es la base práctica para la elaboración del conocimiento teórico.

- Analítico - Sintético, para realizar un estudio a fondo de la bibliografía especializada en cuanto a los SIG y en otros temas esenciales de los parques eólicos, con el fin de identificar elementos claves que contribuyen a la solución del problema planteado. Este método permitirá entender la problemática.

- Histórico - Lógico, para analizar la evolución del proceso de gestión de la información de los parques eólicos, explicar la historia y la lógica interna de su desarrollo.

Método empírico: es el nivel de conocimiento que se obtiene sobre el objeto, a partir de la experiencia, mediante el pensamiento abstracto.

- Entrevista no estructurada, para definir los requisitos funcionales que debe cumplir el SIG, actividad que constituye el cimiento para el diseño y la implementación. La entrevista fue realizada a especialistas del Departamento de Fuentes Renovables de Energía de INEL.

La investigación está estructurada en tres capítulos:

Capítulo # 1 : Fundamentación teórica asociada a la gestión de la información de los parques eólicos.

Definición de conceptos relacionados con los parques eólicos. Síntesis de las posibles soluciones existentes que podrían dar respuesta a la problemática. Descripción de las herramientas definidas junto a la metodología, necesarias para la construcción de la solución propuesta.

Capítulo #2: Diseño e implementación de la solución propuesta.

Se realiza la especificación de los requisitos de software. Se describe la arquitectura, los patrones arquitectónicos y de diseño a emplear. Se incluyen el modelo de datos, los diagramas de despliegue y de componentes. Además, se lleva a cabo la creación de los artefactos ingenieriles fundamentales relacionados con el diseño.

Capítulo #3: Prueba del sistema desarrollado.

Se detalla el flujo de trabajo de pruebas, con la descripción de los casos de prueba, los resultados y no conformidades obtenidas luego de la validación de la solución propuesta.

CAPÍTULO 1. Fundamentación teórica asociada a la gestión de la información de los parques eólicos.

1.1 Introducción.

En la actualidad son numerosos los proyectos que toman como base de su análisis los Sistemas de Información. Muchos de ellos buscan la aplicación de los Sistemas de Información para la resolución de problemas de localización o de evaluación de recursos. Un Sistema de Información Geográfica (SIG), permite a los usuarios representar de forma eficiente, información geográfica referenciada asociada a un territorio. En el presente capítulo, se definen los conceptos más importantes relacionados con dichos sistemas orientados al trabajo con los parques eólicos, junto a las herramientas y tecnologías empleadas en el desarrollo de la solución. A su vez, se argumenta sobre posibles soluciones existentes en Cuba y el mundo, y la importancia de la gestión de la información de campos eólicos en el país.

1.2 Sistemas de Información Geográfica.

Un SIG es un sistema computarizado que permite hacer un análisis de un territorio determinado y manejar datos georeferenciados: entrada de datos, administración de datos (almacenaje y recuperación), manipulación, análisis y salida de datos, apoyando la toma de decisiones, siendo aplicables a cualquier proyecto que requiera trabajo espacial.

1.2.1 Componentes de un SIG.

Los elementos que conforman un SIG, según (Olaya, 2011), son los siguientes:

Equipos (Hardware): Equipo tecnológico en donde opera el SIG.

Programas (Software): Brindan las herramientas que posibilitan la entrada y manipulación de los datos, el análisis, la visualización, las búsquedas y la administración de las bases de datos. Entre los elementos de un SIG se encuentran:

- Sistema gestor de base de datos.
- Una interfaz gráfica de usuarios para el fácil acceso a las herramientas.
- Herramientas para captura y manejo de información geográfica.
- Herramientas para soporte de consultas, análisis y visualización de datos geográficos.

Datos: Son elementos clave de un SIG, pues sin ellos el resto de los componentes no tienen utilidad alguna.

Recurso humano: Para el correcto funcionamiento de los SIG se necesita personal capacitado para operar, desarrollar y administrar el sistema. Los usuarios de SIG varían desde especialistas técnicos, que diseñan y mantienen el sistema, hasta aquellos que lo utilizan para ayudar a realizar sus tareas diarias.

Procedimientos: Los SIG deben operar acorde con un plan bien diseñado, con reglas claras del negocio, que son los modelos y las prácticas operativas características de cada organización.

1.2.2 Conceptos asociados a los SIG.

En busca de lograr un mejor entendimiento de los SIG en la investigación, se hace necesario definir un conjunto de conceptos relacionados con el tema, los cuales se muestran a continuación:

Georreferencia: "*Consiste en ubicar un objeto en el espacio tridimensional con respecto a la tierra utilizando un sistema de coordenadas*." (Scribd inc, 2017)

Cartografía: "*El término cartografía se refiere al ejercicio de hacer mapas. La palabra es una mezcla muy interesante del francés y del griego, Carte es una palabra francesa que significa mapa, mientras que grafía es de origen griego y se refiere a escritura."* (Biblioteca virtual- BR, 2014)

Mapa: *"Representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, que por lo general suele ser plana, aunque también puede ser esférica como en el caso de los globos terráqueos"* (Hernández Barrios, 2012).

Zoom: ***"****Objetivo fotográfico o cinematográfico que mantiene enfocada la imagen al variar la distancia focal"* (The free dicctionary, 2014).

1.3 La gestión de la información de parques eólicos.

La gestión de la información de parques eólicos está definida por un conjunto de acciones u operaciones orientadas a la ejecución de cualquier tipo de actividad que requiera: planificación, desarrollo, implementación y control. La gestión de dicha información busca la rentabilidad y el aprovechamiento de la información; así como mejorar la productividad y competitividad de las entidades relacionadas con el desarrollo de los campos eólico.

1.4 Conceptos asociados a la gestión de la información de los campos eólicos.

1.4.1 Aerogenerador.

Un aerogenerador o turbina eólica es un equipo que convierte la energía cinética del viento en energía eléctrica sin usar combustible, pasando por el estado intermedio de conversión a energía mecánica de rotación a través de las palas. (Fuentenebro, 2014) Las turbinas son clasificadas en vertical u horizontal por la posición de su eje; a sotavento o barlovento por la posición del rotor; por el número de palas, pueden

ser de una, dos, tres palas o multipalas en el caso de una máquina de arrastre; y por el tipo de torre pueden ser tubos de acero, hormigón reforzado o celosía y en caso de emplear angulares, son tubulares o de perfiles de acero. Los aerogeneradores, pueden variar según las especificaciones de los proveedores, pero todos presentan los mismos componentes principales.

1.4.2 Rotor.

El rotor es el motor de la turbina, formado por todo el conjunto de elementos de la misma que gira por delante y fuera de la góndola. (Figueredo, 2013).

1.4.3 Góndola.

La góndola es el componente que sirve de alojamiento para los elementos mecánicos y eléctricos del aerogenerador. Contiene partes claves como el multiplicador y el generador eléctrico, el tren de potencia. (Fuentenebro, 2014)

1.4.4 Torre.

La torre es el elemento que eleva el aerogenerador para evitar las bajas velocidades del viento junto a la superficie del terreno, sitúa el generador a una mayor altura, donde los vientos son de mayor intensidad, para permitir el giro de las palas y transmitir las cargas del equipo al suelo. (Fuentenebro, 2014)

1.4.5 Generador.

El generador es la parte del aerogenerador que convierte la energía en electricidad. Existen diferentes tipos dependiendo del diseño del aerogenerador. Pueden ser síncronos o asíncronos, jaula de ardilla o doblemente alimentados, con excitación o con imanes permanentes. (Fuentenebro, 2014)

1.4.6 Palas.

Las palas son las encargadas de capturar el viento y transmitir esa potencia hacia el buje. Su eje de giro suele estar inclinado algunos grados sobre la horizontal, con el objeto de alejar las palas de la torre. (Fuentenebro, 2014)

1.4.7 Parque eólico.

Un parque o campo eólico, es una agrupación de aerogeneradores que transforman la energía eólica en energía eléctrica; se pueden situar en la tierra o en el mar, siendo los primeros los más habituales. Los parques eólicos proporcionan diferente cantidad de energía dependiendo de las características del diseño de las turbinas, las características de la zona donde se encuentran emplazados o la forma en la que se encuentre diseñado.

1.4.8 Zona de desarrollo.

Una zona de desarrollo, es un territorio que luego de un estudio investigativo sobre sus características geográficas pasa a ser un campo eólico o es clasificado a distintos niveles según su potencial, para ser uno. Las zonas de desarrollo se clasifican en: buenas para el desarrollo, posibles para confirmar, con menos perspectiva y que pueden rechazarse. Las zonas se evalúan en dichas clasificaciones según los criterios de evaluación de: velocidad media del viento, densidad de potencia media del viento, factor de capacidad de las turbinas y el período de medición. Para seleccionar las zonas se desarrollan tres fases; la fase de exploración, donde se hace la prospección eólica general; la fase de planificación, caracterizada por una evaluación detallada del emplazamiento y la fase de operación especializada por la predicción de la energía eólica producida.

1.5 Importancia de la gestión de la información de la energía eólica en el mundo y Cuba.

La gestión de la información de la energía eólica posee gran importancia para las empresas que lo desarrollan, pues permite que estas tengan acceso a información que luego de ser procesada es utilizada para realizar el seguimiento y análisis de los parámetros de funcionamiento de los distintos componentes del parque eólico. Permiten la toma de decisiones en función de los resultados, con el objetivo de mejorar el comportamiento de la instalación y una mayor rentabilidad de la explotación.

1.5.1 Proceso de gestión de energía eólica en Cuba.

En Cuba, el proceso de gestión de energía eólica brinda información sobre los sitios naturales donde se combinan las características del viento (como resultado del calentamiento solar), la cobertura vegetal (franjas de bosques, praderas) y las propiedades de las pendientes que aceleran o no el flujo del viento superficial en la capa de 10 a 20 m de altitud. Permite obtener conocimientos sobre las ventajas y desventajas que presenta el país para el desarrollo del sector eólico y representa una ventaja para las organizaciones en función de lograr la mejora de la rentabilidad y el aprovechamiento de la información que generan para perfeccionar el negocio y el entorno productivo.

Para apoyar dicho proceso se han desarrollado alternativas y sistemas con los cuales se pretendía brindar servicio a las empresas y a los expertos en el tema, gestionando grandes volúmenes de información y asegurando la persistencia de los datos para transmitir la información. Entre los sistemas desarrollados se encuentra el Mapa de Potencial Eólico de Cuba, en donde se evalúa el potencial de la energía eólica, así como los posibles sitios en los cuales es factible el emplazamiento de los parques eólicos a mediana y gran escala que pueden tributar al Sistema Electroenergético Nacional (SEN). Además se desarrolló el Sistema de Gestión de Información de los parámetros del viento, software multiplataforma para la gestión de la información de los parámetros de viento en los parques eólicos que permite la automatización de los

procesos de medición en las estaciones meteorológicas instaladas en los parques y el Sistema de Información Geográfica de Fuentes Renovables de Energía (SIGFRE) que constituye un sistema informático que agrupa diferentes resultados, para diferentes análisis como: conexión de los sistemas renovables a la red eléctrica o estudios de sensibilidad espacial, entre otros. Dichos estudios han estado encaminados al apoyo del proceso de toma de decisiones, por las personas facultadas y dirigidos a lograr el paradigma de la sostenibilidad energética. (Gámez, 2012)

Aunque los sistemas y alternativas anteriores apoyan el estudio y desarrollo de la energía eólica, presentan características que resultan ser limitantes para el cumplimiento de la problemática que hoy presenta INEL; sea por la falta de información, el no mantenimiento del sistema o la no preparación de nuevos expertos. Se realizó, como consecuencia, un estudio de las aplicaciones existentes en el mundo para determinar si pueden ser adaptables a las condiciones existentes en el país.

1.5.2 Sistemas existentes en el mundo.

- *Eólica.*

Eólica es un sistema de gestión, desarrollado por la empresa de ingeniería Davisa, que busca las mejores localizaciones para un parque eólico y optimiza los implantados. Este sistema de gestión, basado en la combinación de varias herramientas informáticas y aprovechando los Sistemas de Información Geográfica (SIG), permite evaluar las zonas óptimas para implantar un parque eólico, conocer información sobre la gestión de permisos, realizar simulaciones y hasta optimizar los recursos de los ya operativos, con sistemas de repotenciación. (Eólica, 2012)

- *El Sistema de Información Geográfica para las Energías Renovables (SIGER) en México.*

El SIGER es una herramienta de apoyo para la toma de decisiones en el área del aprovechamiento de las energías renovables. Por ello, la información más importante con la que cuenta son los mapas de la distribución de los potenciales de las energías renovables. También posee información de utilidad para la determinación de estos potenciales y la determinación de costos de generación eléctrica o de viabilidad técnica de proyectos. La información contenida en el SIGER se encuentra estructurada en tres grandes grupos: energías renovables, cartografía general y cartografía derivada. Dentro del primer grupo se encuentran los mapas del potencial de las energías renovables; en el segundo están los mapas relativos a la geografía física, geografía socioeconómica e infraestructura, y en el tercero se encuentra lo que corresponde a los mapas derivados, obtenidos de los dos primeros. El SIGER es un sistema que cuenta con información tanto de los potenciales de las energías renovables, como de aquellos elementos fundamentales para la evaluación de los recursos y para la determinación de la viabilidad de proyectos en algunas regiones de la República Mexicana. (Miranda, 2012)

- *Explorador de Energía Eólica.*

El Explorador de Energía Eólica es una herramienta en línea, desarrollada en la Universidad de Chile, destinada al análisis del recurso eólico que permite, de manera gráfica y cómoda, realizar una evaluación preliminar del potencial energético sobre cualquier sitio definido por el usuario. Forma parte de un avanzado sistema computacional que simula las condiciones del viento, cuyo componente principal es el modelo WRF (Weather Research and Forecasting), un modelo avanzado, ampliamente utilizado en el mundo para el estudio del potencial eólico. Es importante destacar además que la herramienta asegura un acceso completo, libre y gratuito. El Explorador Eólico no solo posee información atmosférica de mayor calidad, sino que además ofrece una gran cantidad de recursos y herramientas del más alto nivel destinadas a la visualización y análisis detallado del recurso, las que en su conjunto permiten que el usuario lleve a cabo un ejercicio de prospección energética del más alto nivel, y por medio de un navegador web estándar, evitando así la necesidad de programar rutinas típicas o el uso de software propietario. (Matemáticas, 2012)

- *MODERGIS:*

Fue desarrollado por la Universidad Nacional de Colombia y contó con el apoyo del Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT). El objetivo del sistema es determinar la potencialidad de los recursos energéticos renovables, simulando escenarios energéticos que involucra criterios de desarrollo sostenible, brinda seguridad en el suministro y ofrece elementos de juicio en la toma de decisiones sobre la participación de fuentes renovables en la matriz energética, mediante una plataforma de planificación integrada de todas las FRE. La concepción de MODERGIS se convirtió en un instrumento de gran ayuda a la planificación, al integrar en el SIG el modelo de prospección de oferta y demanda de energía, herramientas de análisis multicriterio y multiobjetivo, buscando un mismo objetivo común. Es una aplicación que puede ser utilizada para realizar simulaciones agregadas de todas las fuentes energéticas en el orden nacional (caso Colombia). (Gámez, 2012)

Los sistemas expuestos presentan características que resultan ser limitantes para dar solución a la problemática de INEL. En el caso del sistema Eólica, es propietario y exige autenticación para acceder a su página web; además al igual que ocurre con el MODERGIS, no es adaptable a las características deseadas, porque solo tienen uso en el territorio donde fue desarrollado. El sistema SIGER, presenta un gran vacío en cuanto a la información sobre energía eólica pues no contiene mapas de este recurso y de otros como la biomasa. En el caso particular del SIGFRE, se encuentra offline a causa de la pérdida de los especialistas y desarrolladores que trabajaron en él, y la no preparación de nuevos expertos para darle mantenimiento al sistema y continuar con su desarrollo. Las restricciones mencionadas muestran la imposibilidad de tomarlas como referentes para apoyar el desarrollo de una posible solución informática, y acentúan la necesidad de desarrollar un sistema que si responda a la problemática.

1.6 Metodología seleccionada para desarrollar la solución.

Una metodología de desarrollo de software tiene entre sus objetivos guiar el proceso de desarrollo del software que se produce. Al no existir una metodología universal, toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable (Sánches, 2015). Se trabaja con la metodología AUP-UCI, ya que es la establecida para los proyectos productivos de la universidad.

Según (Sánches, 2015) la metodología AUP-UCI presenta 3 fases, donde en una de ellas se unifican 3 de las fases de AUP. Las fases son:

***Inicio*:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto.

***Ejecución*:** Se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura.

***Cierre*:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Según (Sánches, 2015) la metodología AUP-UCI presenta 7 disciplinas:

***Modelado de negocio*:** Destinada a comprender los procesos de negocio de una organización.

***Requisitos*:** Desarrolla un modelo del sistema que se va a construir.

***Análisis y diseño*:** Se modela el sistema y su forma para que soporte todos los requisitos. Si se considera necesario, permite refinar y estructurar los requisitos para comprensión más precisa.

***Implementación*:** Se construye el sistema.

***Pruebas internas*:** Se verifica el resultado de la implementación probando cada construcción.

***Pruebas de liberación*:** Diseñadas y ejecutadas por una entidad certificadora de calidad externa.

***Pruebas de aceptación*:** Prueba final antes del despliegue del sistema. Se verifica que el software está listo y puede ser usado por usuarios finales.

En el caso específico de la disciplina de requisitos existen 4 escenarios que permiten modelar el sistema en los proyectos. Para el SIG_GIPEC, se trabaja con el escenario 4 que describe historia de usuarios (HU). El cual se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir

los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información. (Sánches, 2015)

1.7 Herramientas propuestas para dar solución al problema.

El perfeccionamiento de las tecnologías de la información es cada vez más elevado en la actualidad, permitiendo el desarrollo de diversas herramientas de software para el progreso de la aplicación informática. Por tal motivo, se hace indispensable seleccionar de forma objetiva las herramientas a utilizar, pues determinará el desarrollo de un buen producto informático. A continuación se analizan diversos métodos y herramientas.

1.7.1 Lenguaje de Modelado UML 2.1

Un lenguaje de modelado consta de vistas, diagramas, elementos de modelo, símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. En todas las disciplinas de la Ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de "algo". Ese "algo" puede existir, estar en un estado de desarrollo o estar todavía en un estado de planeación. Para realizar los modelos del sistema propuesto se hará uso de UML[1], porque permite "visualizar, especificar, construir y documentar los artefactos generados en el desarrollo de un sistema" (Jacobson, y otros, 2012). La visualización, especificación, construcción y documentación de los artefactos generados se realiza a través de diagramas, así como las relaciones entre los diferentes componentes y objetos.

Permite la modificación de todos sus miembros mediante estereotipos y restricciones. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo y se usa solo para modelar sistemas que usan tecnología orientada a objetos. Además, es de fácil entendimiento para todo tipo de roles.

- *Herramienta CASE. Visual Paradigm for UML 8*

Las herramientas de CASE[2] son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste del mismo en términos de tiempo y dinero.

[1] UML: *Lenguaje Unificado de Modelado, UML por sus siglas en inglés.*

[2] *CASE: Ingeniería de Software Asistida por Computadora, CASE por sus siglas en inglés.*

Para apoyar el modelado de los diagramas se hará uso de la herramienta CASE *Visual Paradigm for UML 8*, la cual soporta hasta la versión 2.1 de UML y permite modelar los procesos del negocio, la base de datos y las clases del sistema de manera visual. También soporta patrones de diseño para lograr mejores prácticas y permite la conexión a repositorios como: el CVS [3] y el Subversion.

Esta herramienta presenta gran usabilidad, pues los diagramas se agrupan por categorías, permitiendo al usuario una rápida localización de la información y configuración de los estilos y formatos de los diagramas. Incorpora imágenes y estereotipos, así como la exportación de los diagramas en formato de imagen, además es una herramienta multiplataforma. (VisualParadigm, 1999)

1.7.2 Lenguajes de programación del lado del servidor

- *PHP 5*

PHP[4] es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML" (The PHP Group, 2017).

Es un lenguaje del lado del servidor, por lo que su código se ejecuta en el servidor y el cliente solo recibe una página con el código HTML resultante de la ejecución PHP independientemente del código origen en el servidor. Las páginas que se ejecuten tendrán acceso a bases de datos, conexiones de red y otras funcionalidades para crear la página resultante.

Existe una amplia comunidad de programadores que brinda soporte al lenguaje, lo que posibilita que los fallos de funcionamiento se encuentren y reparen rápidamente. Es simple comparado con otros lenguajes, facilitando el proceso de actualización y/o migración de código, además soporta conexiones a una gran cantidad de bases de datos como: PostgreSQL, MySQL, Oracle, entre otras, y ofrece una solución simple y universal a la paginación dinámica. (The PHP Group, 2017)

Se escoge PHP5 como lenguaje de programación debido a que, complementando lo planteado anteriormente, proporciona una gran mezcla entre rendimiento y flexibilidad en la realización de páginas dinámicas, además el equipo de desarrollo cuenta con experiencia en la implementación de aplicaciones web usando PHP como lenguaje.

---

[3] *CVS: Concurrent Versions/Versioning System o Versión Concurrente.*

[4] *PHP: Acrónimo recursivo de Hypertext Preprocessor o Pre-procesador de hiper-texto.*

- *PHP MapScript*

Es un módulo para PHP que permite acceder a la API[5] de MapServer. Carga las capas configuradas en un mapfile al inicializar y permite modificar, cambiar e incluso agregar más capas.

Las librerías MapScript de PHP constituyen la vía de comunicación de las aplicaciones SIG con el servidor de mapas MapServer y por otra parte rompen en cierta medida ésta rigidez de la representación de mapas a través de los ficheros "Mymapfile.map" permitiendo modificar el mapfile en tiempo de ejecución, a los cuales se les conoce como mapfile dinámicos, facilitando de esta forma la creación de aplicaciones con un grado de personalización mayor. (Jaime M. Tan Nozawa, 2007)

1.7.3 Lenguaje de programación del lado del cliente

Usado principalmente en su forma del lado del cliente, *JavaScrip*t es un lenguaje interpretado, es decir que su código no se compila, sino se ejecuta a través de un intérprete. Este lenguaje es utilizado mayormente en páginas web. Todos los navegadores modernos interpretan el código JavaScript de las páginas web, el cual tiene integrado para tal propósito una implementación del Modelo de Objetos de Documentos (Mozilla Developer Network, 2005).

Es un lenguaje débilmente tipado, orientado a objetos y dinámico. Responde a eventos en tiempo real y cambia el aspecto de la página a gusto del usuario. Incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

Teniendo en cuenta lo anterior, se selecciona a JavaScript como el lenguaje de programación del lado del cliente ya que garantiza flexibilidad y rapidez a la aplicación, independientemente del navegador que utilice el cliente.

- *ExtJS 3.0*

JS Ext o ExtJS es una biblioteca JavaScript que incluye: componentes UI[6] del alto rendimiento y personalizables, modelo de componentes extensibles, una API fácil de usar y licencias *OpenSource*[7] y comerciales. Permite crear aplicaciones complejas utilizando componentes predefinidos, así como un manejador de *layouts*, que le posibilita proveer una experiencia consistente sobre cualquier navegador, evitando validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.). La ventana flotante que provee JS Ext se considera excelente por la forma en la que funciona. Al moverla o

[5] *API: Interfaz de Programación de Aplicaciones o Application Programming Interface.*

[6] *UI: Interfaz de usuario o user interface.*

[7] *OpenSource: Software de código abierto en inglés open source software u OSS.*

redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido. Además, brinda los siguientes beneficios:

- Balance entre Cliente – Servidor: La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona: En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente lo note.
- 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 (Mozilla Developer Network, 2005).

1.7.4 Sistema gestor de Base de Datos. PostgreSQL 9.3

Un Sistema Gestor de Base de Datos es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, y controlar la entrada y salida de datos de la base de datos, manteniendo la integridad de la misma. La importancia de su uso es que proporciona a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos.

Durante el transcurso de la investigación, se emplea PostgreSQL, el cual es un sistema de gestión de bases de datos objeto-relacional. Se encuentra distribuido bajo licencia BSD y tiene su código fuente disponible libremente. Es uno de los sistemas de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Esto ofrece como ventaja que un fallo en uno de los procesos no afecte al resto, garantizando así que el sistema continúe funcionando.

Entre sus principales características se encuentra la creación de copias de seguridad mientras se usa la base de datos, lo que permite salvar las operaciones realizadas cuando ocurra un suceso inesperado. Además, utiliza para el control de concurrencia un modelo multiversión, creando una “copia” de la base de datos para cada usuario. Los cambios que se realicen solo se reflejarán en la base de datos del sistema cuando hayan finalizado todas las operaciones, actualizando posteriormente todas las “copias” a todos los usuarios conectados. De esta forma se protegen las transacciones y permite que múltiples usuarios trabajen simultáneamente sobre una misma base de datos sin que ocurran interbloqueos en la ejecución de las consultas (PostgreSQL, 2009).

Debido a sus características y potencialidades, se selecciona a PostgreSQL en su versión 9.3 como el sistema de gestión de base de datos para el desarrollo de la aplicación.

- *PostGIS: la extensión geográfica del gestor seleccionado*

Es una extensión de datos espaciales para PostgreSQL, que añade soporte a objetos geográficos permitiendo realizar consultas de ubicación para ser ejecutadas en SQL.

PostGIS 2.1.3: Es una versión estable y liberada de PostGIS, extensión de PostgreSQL que la convierte en una Base de Datos Espacial. Posee soporte de datos raster y vectoriales. Es compatible con el servidor de mapas MapServer. (MappingGIS, 2015)

1.7.5 Servidor de aplicaciones Apache 2.4.7

Apache es un servidor web multiplataforma de código abierto. Es modular (basado en módulos), donde cada módulo ofrece un grupo de funcionalidades específicas al servidor. Es uno de los servidores web más utilizados en Internet, lo que facilita el acceso a la documentación. Provee un alto nivel de seguridad y eficiencia, permitiendo además el uso de una versión local, la cual hace posible que el servidor actúe como servidor y cliente al mismo tiempo, creando así la posibilidad de pre visualizar y probar el código mientras este es desarrollado (Apache, 2013). A partir de estas características, se decide utilizar Apache versión 2.4.7 como servidor web.

1.7.6 Servidor de mapas MapServer 6.4.1

MapServer es un entorno de desarrollo o plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas en Internet/Intranet. Permite visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet MapServer. Ofrece la posibilidad de ser utilizado como servidor de mapas de terceros programas y admite múltiples formatos de datos vectoriales, características que hacen de MapServer una herramienta potente (Sphinx, 2014).

Por otra parte, esta es la herramienta utilizada en el proyecto donde radica el equipo de desarrollo, por lo que se cuenta con abundante bibliografía y con personal experimentado en su uso. Todo esto hace de MapServer el servidor de mapas ideal para el desarrollo de la aplicación que brinda solución al problema de la investigación, por tal motivo se decide emplear la versión 6.4.1 de esta plataforma.

Entre las características más relevantes de este servidor se destaca la presencia de soporte para datos ráster (TIFF/GeoTIFF, EPPL7) y vectoriales (ESRI shapefiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL) y la posibilidad de integración con las bibliotecas GDAL[8] y OGR[9], para aumentar el número de

[8] *GDAL: Geospatial Data Abstraction Library o Biblioteca de Abstracción de Datos Geoespaciales.*

[9] *OGR: Simple Features Library o Biblioteca de Funciones Sencillas.*

formatos soportados. Otra característica importante es su capacidad de trabajar de forma interoperable con diversos SIG y aplicaciones que siguen las especificaciones establecidas por el OGC. Esta última característica es la que permite la publicación de un fichero de mapa como servicio Web de mapas, posibilitando que puedan establecerse mecanismos de caché entre las aplicaciones clientes y el servidor de mapas.

1.7.7 Plataforma GeneSIG 2.0

La plataforma GeneSIG es una herramienta que permite realizar representaciones y análisis de información referenciada geográficamente (Varen Caballero, 2010). Posee una estructura basada en *plugins*, lo que la convierte en una plataforma con un alto grado de interoperabilidad debido a que permite agregar o quitar componentes de manera sencilla. De igual modo permite separar los servidores de bases de datos y web en dos estaciones de trabajo diferentes, balanceando la carga del sistema, aumentando su disponibilidad y disminuyendo la posibilidad de fallas (Castañeda Rodríguez, 2014).

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

Cumple y respeta la estructuración y el diseño propuestos por CartoWeb, pero solo utiliza los paquetes a los cuales le realizará cambios o aportes funcionales. Presenta también un abanico bastante completo de características propias de un geoportal, con posibilidad de ir añadiendo o desarrollando nuevos plugins. Es precisamente a través de estos plugins agregados de forma convencional, que GeneSIG posee un amplio conjunto de funcionalidades, que actúan como herramientas de la misma plataforma y le brindan la posibilidad de ser altamente modular y escalable.

Entre sus principales módulos se encuentran:

Módulo de Navegación: Se encarga de gestionar toda la interacción del usuario con la interfaz visual donde se encuentra el mapa y garantiza que este pueda realizar las operaciones de movimiento, acercamiento y alejamiento en sus diferentes variantes.

Módulo de Selección: Ofrece la posibilidad de selección de objetos geográficos dentro de las capas seleccionables definidas y realizar operaciones de consulta o persistencia de selección.

Módulo de Consulta Espacial: Permite consultar espacialmente objetos puntuales o los determinados por un área que defina el usuario, rectangular, circular o poligonal.

Módulo de Análisis: Ofrece la posibilidad de realizar análisis espacial sobre el mapa como cálculo de distancias, superficies y localizaciones.

Módulo de Configuración del Mapa: Permite la configuración de la aplicación para el manejo de los datos del mapa, unidades de medidas, proyección, entre otros.

Módulo de Impresión: Ofrece la posibilidad de imprimir, la información que defina el usuario en formato Excel.

Módulo de Catálogo: Posibilita la configuración de la representación del mapa en cuanto a estilos y simbología desde una interfaz amigable.

Módulo de Servicios: Realiza operaciones relacionadas con la gestión de las capas, como selección y agregado de nuevas capas en diferentes formatos y conexión a servicios de mapas.

Módulo de Edición: Realiza operaciones de edición que permiten a los usuarios insertar nuevos objetos geográficos a capas temporales.

Módulo de Ayuda: Relaciona opciones de apoyo al usuario para el trabajo con la plataforma.

La arquitectura de la plataforma GeneSIG es una arquitectura distribuida, empleando como base cartográfica una información certificada por especialistas que laboran en su desarrollo, y sobre ella un conjunto de objetos representados geoespacialmente que contienen información asociada. El sistema tiene una arquitectura cliente-servidor, orientada a objetos y basada en componentes.

1.7.8 Entorno de Desarrollo Integrado (IDE). NetBeans 8.1

Netbeans es un Entorno de Desarrollo Integrado (IDE) que permite crear aplicaciones de escritorio, aplicaciones web, entre otras. Es una aplicación gratuita y sin restricciones de uso. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándoles nuevos módulos. Debido a que pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (Moreta Fernández, 2013)

Las principales características de NetBeans son:

- Entorno de desarrollo multiplataforma, multilenguaje e intuitivo tanto para usuarios novatos como para profesionales.
- Es de código abierto.

- Desarrollado por módulos. Brinda la posibilidad de agregar nuevos módulos para aumentar su funcionalidad.
- Cuenta con una amplia documentación y una gran comunidad de usuarios.
- Admite distintos tipos de lenguaje como PHP, C++ y Java.

Teniendo en cuenta las características mencionadas se seleccionó NetBeans 8.1 como IDE de desarrollo para la implementación del sistema utilizando el lenguaje PHP.

1.8 Conclusiones del capítulo.

Con la caracterización de los SIG se pudo demostrar la importancia de los mismos para lograr una mejor visión del análisis de la información geográfica, ya que son usados como fuente principal para la gestión y manipulación de datos geográficos. El estudio de soluciones existentes, sirvió de apoyo a la investigación para lograr un mejor entendimiento del sistema a desarrollar y al análisis detallado de la situación problemática antes planteada. La definición de la metodología garantizó la generación de la documentación necesaria para incorporar futuras funcionalidades al sistema o realizar una nueva versión. Las herramientas y tecnologías, definieron un entorno técnico con el cual se realizará el desarrollo de la solución.

CAPÍTULO 2. Diseño e implementación de la solución propuesta.

2.1 Introducción

Durante el desarrollo de este capítulo se aborda todo lo referente a la disciplina de requisitos perteneciente a la metodología AUP-UCI, en el cual se analizan todas las entidades y conceptos presentes en el entorno donde se aplica el sistema. Se realiza el levantamiento de requisitos funcionales y no funcionales, los mismos guiarán el proceso de desarrollo del futuro producto hasta su implementación. Además, se crean los diagramas correspondientes al modelo de diseño, que son de vital importancia para la implementación del sistema. Concluyendo con la descripción de los elementos de la arquitectura y los patrones arquitectónicos y de diseño a emplear.

2.2 Especificación de requisitos.

El levantamiento de requisitos se realiza con el objetivo de comprender mejor el problema que se necesita solucionar, existen los Requisitos Funcionales (RF) los cuales especifican los detalles más relevantes que la aplicación debe presentar y los Requisitos No Funcionales (RNF) que son las cualidades que la misma debe poseer para un mejor funcionamiento.

2.2.1 Requisitos Funcionales:

Los Requerimientos Funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. Los RF describen lo que el sistema debe hacer (Sommerville, 2005). A continuación, se describen los requisitos funcionales de la solución.

Navegación:

RF 1: Acercar determinada región del mapa

Esta funcionalidad permite aumentar el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de acercar. Para realizar esta funcionalidad se requiere que el usuario seleccione en el mapa el punto o la región que desea acercar y el sistema mostrará dicha región a un mayor nivel de visualización.

RF 2: Alejar determinada región del mapa.

Esta funcionalidad permite disminuir el tamaño del mapa, mostrando en pantalla la región donde el usuario realice la operación de alejar. Una vez disminuido el tamaño se aumenta la escala del mapa. Para realizar esta funcionalidad se requiere que el usuario seleccione en el mapa el punto que desea alejar y el sistema mostrará dicha región a un menor nivel de visualización.

RF 3: Visualizar todo el mapa.

Esta funcionalidad permite visualizar el mapa en su escala máxima. Para realizar esta funcionalidad el usuario debe seleccionar la opción correspondiente.

RF 4: Modificar el centro del mapa.

Esta funcionalidad permite modificar el centro del mapa, sin cambiar la escala del mismo, poniendo como centro el nuevo punto seleccionado por el usuario. Para realizar esta funcionalidad se requiere que el usuario seleccione en el mapa la región que desea centrar y el sistema mostrará dicha región en el centro del mapa.

RF 5: Mover el mapa.

Esta funcionalidad permite al usuario mover el mapa para visualizar otras regiones.

RF 6: Navegar a través del mapa de referencia.

Esta funcionalidad permite al usuario visualizar otras regiones del mapa principal, haciendo uso de un Mapa de Referencia, posicionándose en la zona seleccionada.

Identificación:

RF 7: Realizar identificación puntual.

Esta funcionalidad permite que el usuario pueda consultar la información asociada a los objetos que forman parte de las capas del mapa seleccionadas. Mediante la identificación puntual se obtiene la información de todos los objetos que se encuentran en el punto señalado por el usuario en el mapa. El sistema mostrará los datos asociados a los objetos, en este caso serán las zonas prospecto, los parques eólicos y las turbinas.

RF 8: Realizar identificación rectangular.

Esta funcionalidad permite que el usuario pueda consultar la información asociada a los objetos que forman parte de las capas del mapa seleccionadas. Mediante la identificación rectangular se obtiene la información de todos los objetos que se encuentran dentro del rectángulo dibujado en el mapa. El usuario debe seleccionar la opción correspondiente y luego dibujar un rectángulo que encierre el o los objetos de los que desea obtener la información. El sistema mostrará los datos asociados a las zonas prospecto, los parques eólicos y las turbinas.

RF 9: Realizar identificación poligonal.

Esta funcionalidad permite que el usuario pueda consultar la información asociada a los objetos que forman parte de las capas del mapa seleccionadas. Mediante la identificación poligonal se obtiene la información de todos los objetos que se encuentran dentro del polígono dibujado en el mapa. El usuario debe seleccionar

la opción correspondiente y luego dibujar un polígono que encierre el o los objetos de los que desea obtener la información. El sistema mostrará los datos asociados a las zonas prospecto, los parques eólicos y las turbinas.

Localización:

RF 10: Localizar por elemento (turbina, parque o zona).

Esta funcionalidad permite al usuario localizar sobre el mapa un elemento deseado. Como resultado se mostrará el mapa de forma tal que dicho elemento quede en el centro, con una mejor visualización.

Para esta funcionalidad se requiere como criterio de entrada:

- Tipo de elemento (Formato: Campo de selección, Obligatorio: Sí).
- Valor de filtrado (Formato: Alfanumérico, Obligatorio: Sí).

Visualización:

RF 11: Habilitar las capas del mapa.

Esta funcionalidad permite habilitar las capas del mapa que se encuentren deshabilitadas y que el usuario desea visualizar en la aplicación. Se debe seleccionar en el panel izquierdo en el Control de Capas, la capa que se quiere habilitar y refrescar el mapa.

RF 12: Deshabilitar las capas del mapa.

Esta funcionalidad permite deshabilitar las capas del mapa que se encuentran habilitadas y que el usuario no desea que se muestren en la aplicación. Se debe seleccionar en el panel izquierdo en el Control de Capas, la capa que se quiere deshabilitar y refrescar el mapa.

RF 13: Modificar la escala del mapa.

Esta funcionalidad permite al usuario modificar la escala con la que se visualiza el mapa en la aplicación.

Para esta funcionalidad se requiere como criterio de entrada:

- Escala (Formato: Numérico, Obligatorio: Sí).

El sistema muestra como resultado el mapa en la escala seleccionada.

Impresión:

RF 14: Exportar mapa como imagen.

Esta funcionalidad permite que el usuario pueda exportar el mapa o una región del mismo en formato de imagen. Para ello el usuario deberá seleccionar en la barra de herramientas la opción correspondiente, así como la región que desea exportar. El sistema exportará dicha región como una imagen.

Análisis:

RF 15: Realizar tematización por colores.

Esta funcionalidad le permite al usuario realizar una tematización por colores, a partir de criterios de análisis definidos. Mediante esta tematización el usuario podrá conocer según los colores establecidos las zonas prospecto por su clasificación, los parques eólicos con generación mayor, igual o menor que la media y la cantidad de turbinas según su estado (rota, en reparación o funcional). Para ello se requieren los siguientes criterios de entrada:

- Criterio de análisis (Formato: Campo de selección, Obligatorio: Sí)

  Generación de los parques eólicos.

  Turbinas afectadas.

  Zonas prospecto por clasificación.

RF 16: Medir distancia.

Esta funcionalidad le permite al usuario calcular la distancia entre dos o más puntos a partir del trazado de los mismos con el mouse y visualizar el valor de la distancia entre los dos últimos vértices dibujados y la distancia acumulada (del primer al último vértice) en un panel lateral, además de poder modificar la unidad de distancia si lo desea. Para su desarrollo se requiere el siguiente criterio de entrada:

- Unidad de medida (Formato: Alfabético).

Esta funcionalidad devolverá el valor correspondiente a la distancia según la polilínea trazada:

- Distancia (Formato: Alfanumérico).
- Distancia acumulada (Formato: Alfanumérico).

RF 17: Calcular área y perímetro.

Esta funcionalidad permite que el usuario pueda calcular el área y perímetro de una región a partir del trazado de la misma con el mouse sobre el mapa, visualizando los valores de los mismos, además de poder modificar la unidad del área si lo desea.

Gestión:

RF 18: Registrar la energía generada por las turbinas.

Esta funcionalidad permite que el usuario pueda registrar en el sistema la energía generada por las turbinas en un parque eólico.

RF 19: Modificar la energía generada por las turbinas.

Esta funcionalidad permite que el usuario pueda modificar en el sistema la energía generada por las turbinas en un parque eólico.

RF 20: Adicionar turbina en un parque eólico.

Esta funcionalidad permite que el usuario pueda adicionar en el sistema una turbina en un parque eólico.

RF 21: Eliminar turbinas.

Esta funcionalidad permite que el usuario pueda eliminar en el sistema turbinas.

RF 22: Modificar turbina.

Esta funcionalidad permite que el usuario pueda modificar en el sistema los datos de una turbina.

RF 23: Adicionar parque eólico.

Esta funcionalidad permite que el usuario pueda adicionar en el sistema un parque eólico en una zona prospecto.

RF 24: Eliminar parques eólicos.

Esta funcionalidad permite que el usuario pueda eliminar en el sistema parques eólicos.

RF 25: Modificar parque eólico.

Esta funcionalidad permite que el usuario pueda modificar en el sistema los datos de un parque eólico.

RF 26: Adicionar zona prospecto.

Esta funcionalidad permite que el usuario pueda adicionar en el sistema una zona prospecto.

RF 27: Modificar zona prospecto.

Esta funcionalidad permite que el usuario pueda modificar en el sistema los datos de una zona prospecto.

RF 28: Eliminar zonas prospecto.

Esta funcionalidad permite que el usuario pueda eliminar en el sistema zonas prospecto.

RF 29: Adicionar proveedor

Esta funcionalidad permite que el usuario pueda adicionar un proveedor en el sistema.

RF 30: Modificar proveedor.

Esta funcionalidad permite que el usuario pueda modificar los datos de un proveedor en el sistema.

RF 31: Eliminar proveedor.

Esta funcionalidad permite que el usuario pueda eliminar un proveedor en el sistema.

Reportes:

RF 32: Estadística de los parques eólicos por generación eléctrica.

Esta funcionalidad permite obtener un reporte en formato Excel, donde se muestre un listado con la generación total de los parques eólicos.

RF 33: Generación energética de un parque eólico en un periodo de tiempo determinado.

Esta funcionalidad permite obtener un reporte en formato Excel, donde se muestre un listado de la generación de energía de un parque eólico según el periodo de tiempo seleccionado por el usuario. Para ello se requieren los siguientes criterios de entrada:

- Criterio de análisis (Formato: Campo de selección, Obligatorio: Si)
  Nombre del parque.
- Rango de fecha (Formato: Campo de selección, Obligatorio: Si)

  Fecha inicial.

  Fecha final.

RF 34: Cantidad de turbinas afectadas.

Esta funcionalidad permite obtener un reporte en formato Excel, donde se muestre un listado con las turbinas que se encuentren en estado rota o en reparación.

RF 35: Cantidad de zonas prospecto según clasificación.

Esta funcionalidad permite obtener un reporte en formato Excel, donde se muestre un listado de las zonas prospecto según la clasificación que defina el usuario. Para ello se requieren los siguientes criterios de entrada:

- Criterio de análisis (Formato: Campo de selección, Obligatorio: Sí)
  Clasificación.

Seguridad:

RF 36: Autenticar usuario.

Esta funcionalidad permite al usuario autenticarse para iniciar sesión en el sistema.

RF 37: Cerrar sesión.

Esta funcionalidad permite al usuario cerrar sesión en el sistema.

RF 38: Adicionar usuario.

Esta funcionalidad permite adicionar nuevos usuarios para que puedan acceder al sistema.

RF 39: Eliminar usuario.

Esta funcionalidad permite eliminar los usuarios que se encuentren registrados en el sistema.

RF 40: Modificar usuario.

Esta funcionalidad permite modificar los datos de un usuario registrado en el sistema.

2.2.2 Requisitos No Funcionales

Los Requisitos No Funcionales (RNF) son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los RNF especifican o restringen las propiedades emergentes del sistema (Sommerville, 2005).

Usabilidad

El objetivo de este requisito es definir la usabilidad del sistema, para ello se describen las características del usuario final que usará el sistema, el tipo de aplicación informática, el objetivo que persigue la aplicación y las características de hardware y software requeridas para desplegar y utilizar la aplicación.

RNF 1 Tipo de usuario final

Debido a las acciones que se realizarán en la aplicación deberá ser manejada por usuarios con conocimientos básicos en Informática. En la siguiente tabla se describen las características de los usuarios finales de la aplicación.

Tabla 1: Características de los usuarios finales de la aplicación.

Sexo	Edad	Nivel de escolaridad	Ocupación	Experiencia profesional
F/M	Laboral	Universitario	Especialistas	1 año o más

RNF 2 Tipo de Aplicación Informática

La solución constituye una aplicación Web que permitirá la gestión y representación sobre un mapa de los elementos que maneja el departamento de Recursos Renovables de INEL, así como otras funcionalidades básicas de todo Sistema de Información Geográfica.

RNF 3 Finalidad

La aplicación que se propone tendrá como objetivo fundamental el análisis y representación geoespacial de la información manejada por el departamento de Recursos Renovables de INEL.

RNF 4 Ambiente

Requisitos de Hardware y Software para Servidor Base de Datos y Servidor de Mapas

Hardware:

- Procesador: 3Ghz.
- Memoria RAM: 4Gb.
- Disco Duro: 1Tb.

Software:

- Sistema Operativo: GNU/Linux Ubuntu 14.04.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5.3 configurado con la extensión Pgsql incluida.
- PostgreSQL 9.3 como Sistema Gestor de Base de Datos.
- PostGIS 2.1.3 como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 6.4.1, con extensión PHP MapScript.

Requisitos de Hardware y Software para estaciones clientes.

Hardware:

- Procesador: 1.80 GHz como mínimo.
- Memoria RAM: 2 Gb como mínimo.
- Disco Duro: 150 Gb.
- Tarjeta de red.

Software:

- Un navegador como Mozilla Firefox versión 10 como mínimo u otro que cumpla con los

estándares W3C.

Confiabilidad

RNF 5 Requisito de confiabilidad

El objetivo de este requisito es describir los aspectos que garantizarán la seguridad de la información que se maneja en el sistema propuesto.

- Los datos del sistema no serán modificados por los usuarios no autorizados, debido a que se deniega el acceso a los mismos.
- El sistema deberá ser utilizado por los usuarios que se encuentran registrados en el mismo.
- Los roles definidos en el sistema son: Administrador y Especialista.

RNF 6 Restricciones de diseño

El objetivo de este requisito es definir las restricciones de diseño del sistema a construir.

Los aspectos que a continuación se exponen garantizarán el correcto funcionamiento del software y su escalabilidad en cuanto al diseño de implementación.

- Para la modelación del sistema se utilizará como lenguaje de modelado UML 2.1 y como herramienta el Visual Paradigm for UML 8.0.
- El sistema estará implementado en lenguaje PHP, en su versión 5. Se desarrolla a partir de la plataforma GeneSIG, construida sobre CartoWeb como framework de desarrollo de Sistemas de Información Geográfica.
- Se hace uso de NetBeans 8.1 como entorno de desarrollo integrado.
- Gestor de base de datos PostgreSQL 9.3.
- El sistema se construirá siguiendo la arquitectura que es propuesta por su plataforma base. Los patrones de arquitectura empleados son Basado en componentes y Orientado a objetos.
- El sistema usa para la tematización los colores y rangos establecidos por el usuario y debe mostrar además una leyenda especificando el significado de estos.

RNF 7 Interfaz

El objetivo de este requisito es definir las interfaces que deben ser soportadas por la aplicación.

Interfaces de usuario

- Deben tener un diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.
- El sistema debe mostrar interfaces con íconos intuitivos representando las funcionalidades que se pueden realizar.
- El sistema debe brindar la posibilidad de ocultar los paneles laterales para visualizar el mapa en toda la amplitud de la pantalla.
- El sistema debe tener un paginado para cada listado de elementos, de forma tal que se pueda mostrar solo una cantidad determinada de elementos por página.

2.3 Historia de Usuario.

La Historia de Usuario es el cuarto escenario que presenta la metodología AUP-UCI. Se le aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información; definen descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica, se realiza una por cada funcionalidad del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación. A continuación se muestra la relación de las historias de usuario correspondientes a los requisitos de Realizar identificación puntual y Adicionar zona geográfica (Tablas 1 y 2).

2.3.1 Relación de las Historias de Usuarios.

Tabla 2: HU_ Realizar identificación puntual

Número: 7	**Nombre del requisito:** Realizar identificación puntual.
Programador: Chabelys M. Lestapier Lao Maibol Stable Bientz	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:** 2 días
Riesgo en Desarrollo:	**Tiempo Real:**

• Rotura del host donde esté instalado el servidor de BD. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección del país o de la universidad. • Planificación irreal.	
Descripción: Permitirá al usuario consultar la información asociada a los objetos del mapa. Para ello debe marcar el punto sobre el mapa y se le mostrará la información de los elementos encontrados en dicho punto.	
Observaciones: • El sistema muestra como resultado la información del elemento marcado. • Pueden realizar la funcionalidad todos los roles del sistema de acuerdo a la información a la que puede tener acceso cada uno de ellos.	
Prototipo de interfaz:	

Tabla 3: HU_ Adicionar zona prospecto.

Número: 26	**Nombre del requisito:** Adicionar zona geográfica.
Programador: Chabelys M. Lestapier Lao Maibol Stable Bientz	**Iteración Asignada:** 1
Prioridad: *Alta*	**Tiempo Estimado:** 5 días
Riesgo en Desarrollo:	**Tiempo Real:**

• Rotura del host donde esté instalado el servidor de BD. • Afectaciones al personal de trabajo, debido a orientaciones de la dirección del país o de la universidad. • Planificación irreal.	

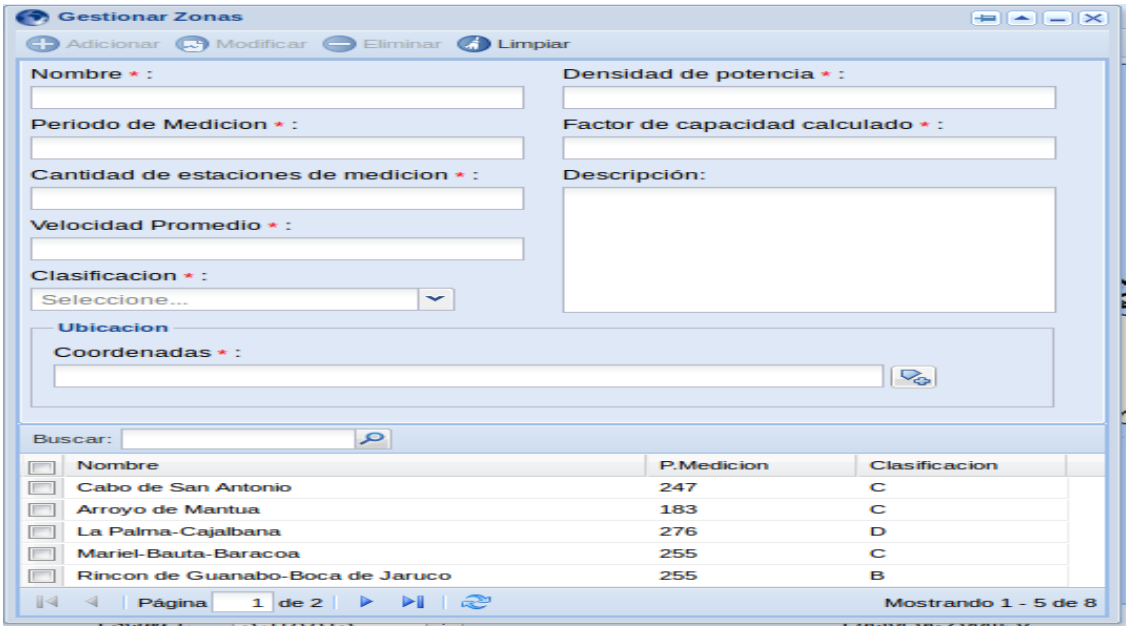
Descripción:

Permitirá adicionar zonas geográficas en el sistema.

Observaciones:

Pueden realizar la funcionalidad todos los roles del sistema.

Prototipo de interfaz:



2.4 Arquitectura de software.

Con el propósito de facilitar la implementación del sistema, se procede a diseñar su arquitectura. *"La arquitectura de software es un conjunto de patrones que proporciona un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema, puesto que establece la estructura, funcionamiento e interacción entre las partes del software"* (La Arquitectura de Software, 2014).

Se le considera un elemento fundamental para la creación de sistemas informáticos, ya que constituye un modelo relativamente pequeño y comprensible de cómo será estructurado el sistema y cómo trabajan juntos sus componentes (Bass, y otros, 2015). Se caracteriza por incluir los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos (Pressman, 2010).

Muchas de las herramientas tecnológicas siguen determinadas arquitecturas, que son transferidas a los sistemas que se desarrollan haciendo uso de ellas. Tal es el caso de la plataforma GeneSIG, la cual emplea la arquitectura orientada a objetos y basada en componentes. Seguidamente se argumenta al respecto.

2.4.1 Patrones arquitectónicos

Patrón de Arquitectura Orientada a Objetos: Esta arquitectura tiene la gran ventaja de descomponer el trabajo en módulos, lo cual disminuye la complejidad. De igual modo fomenta la escalabilidad y la modularidad, a la vez que facilita la reutilización de códigos.

En el caso del producto a desarrollar, la base arquitectónica se modela haciendo uso de este patrón, dado que el *framework* que se emplea para su desarrollo es CartoWeb y este posee dichas características. Se evidencia que existen relaciones de herencia, agregación y composición dentro de diferentes clases del sistema. *"En estudios arquitectónicos sobre estilos, esta arquitectura aparece relativamente relacionada a las arquitecturas basadas en componentes"* (Reynoso, y otros, 2013).

Patrón de arquitectura basada en componentes: Este patrón se centra en el diseño y construcción de sistemas computacionales que utilizan componentes de software reutilizables. Aboga por la utilización de componentes prefabricados, es por ello que una de sus principales características es la reusabilidad. De igual modo, la compatibilidad, la modularidad y la robustez, forman parte de sus peculiaridades. Su premisa es que los componentes cumplan con alta cohesión y bajo acoplamiento (Reynoso, y otros, 2013).

Debido a la filosofía del *framework* utilizado, también se hace uso de este patrón en la solución informática. Se evidencia en la estructura de CartoWeb ya que está compuesto por *plugins* los cuales se dividen en dos

grupos, los *core-plugins* y los *plugins*. Los primeros son de obligatoria presencia ya que son los utilizados por el sistema y los segundos son los creados por los desarrolladores. Esta posibilidad que brinda el *framework* permite que la arquitectura del sistema sea flexible y fácil de personalizar, ya que posibilita agregar o modificar *plugins* sin que se vea afectado el resto de los componentes.

2.4.2 Estilo arquitectónico.

Se selecciona el estilo llamada y retorno, el cual permite obtener una estructura de programa que resulta relativamente fácil de modificar y cambiar de tamaño. Esta familia son los más generalizados en sistemas en gran escala (Reynoso, y otros, 2013).

2.5 Modelo de diseño.

El modelo de diseño, según IBM[10], se basa en describir con mayor detalle, la estructura y el funcionamiento del sistema. En él se representan los componentes de la aplicación y se determina la ubicación y el uso adecuado dentro de la arquitectura general (IBM, 2013).

En el modelo de diseño, los paquetes contienen los elementos de diseño del sistema, tales como clases de diseño, interfaces y subsistemas de diseño, que se desarrollan a partir de las clases de análisis. Estas capas arquitectónicas constituyen la base de una organización y la implementación del sistema.

En resumen, el modelo de diseño sirve para mostrar una vista general del sistema a construir, ayuda a entender, aclarar y transmitir las ideas sobre el código y los requisitos. Describe las relaciones y los comportamientos de los requisitos y las estructuras de un sistema. Es por ello que son ampliamente usados en el proceso del diseño de los sistemas y los componentes que se encargaran del funcionamiento y la relación entre ellos. Entre los artefactos que genera el modelo de diseño se encuentra el Diagrama de Clases del Diseño donde se muestran las clases, interfaces y colaboraciones, así como las relaciones entre ellas.

En la Figura 1 se expone la representación de uno de los diagramas de clases de diseño correspondiente al *plugin "Localizar por elementos"* del sistema. Esta representación contribuye a una correcta implementación del producto. El resto de los diagramas de clases de diseño de la aplicación, se encuentran en el artefacto "GEYSED_SIG_GIPEC", el cual se encuentra asociado a la investigación.

[10] IBM: International Business Machines Corp. es una reconocida empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York.

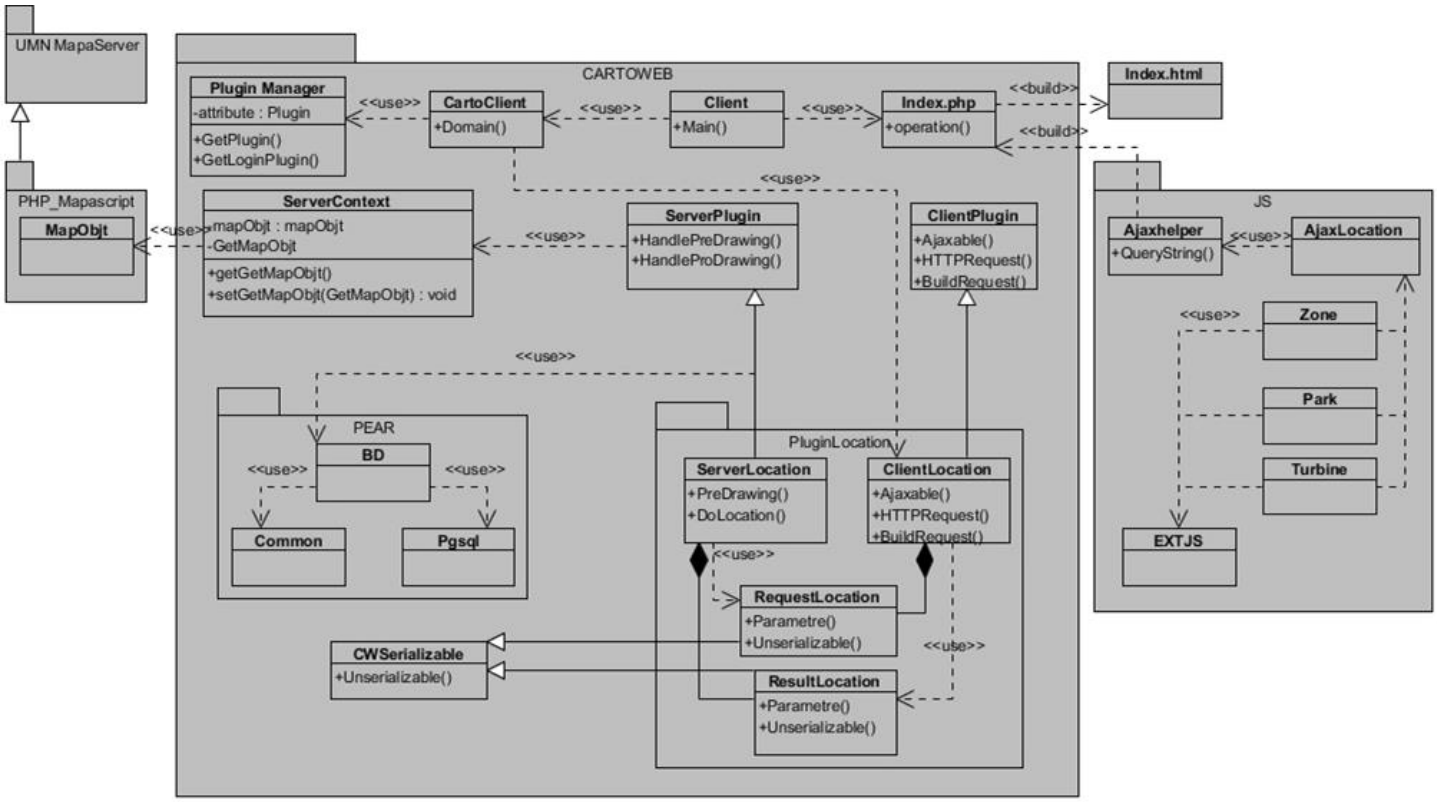


Figura 1: Diagrama de clases de diseño: Localizar por elementos.

Para una mejor comprensión del diagrama, se especifican los elementos que lo componen:

Clase index.php: Recibe las peticiones realizadas por el cliente, gestiona las mismas y manda a construir la ClientPage.

Clase Client: Contiene todos los archivos específicos de PHP del lado de CartoClient y permite la interacción entre la index.php y la CartoClient.

Clase CartoClient: Integra y recoge todos los datos y funciones realizadas por cada una de las .js. En ella se define una serie de variables globales que emplea la aplicación.

Clase PluginManager: Se utiliza para gestionar la base de plugins.

Clase ClientPlugin: Contiene las interfaces necesarias para los plugins del lado del cliente.

Clase ServerPlugin: Proporciona la base de herramientas para el desarrollo de plugins.

Clase ServerContext: Es la contenedora de la información común que se utiliza por la parte cliente y la servidora, empleando la información seleccionada como un objeto para un fácil manejo de los datos.

Clase MapObj: Es donde se definen los métodos, funciones y lenguaje para el intercambio de datos con el servidor de mapa MapServer.

Clase BD: Se encarga de establecer la conexión con el servidor de base de datos para procesar los objetos a adicionar, modificar y eliminar.

Clase Common: Es la encargada de administrar las conexiones a la base de datos para ejecutar las consultas a la misma satisfactoriamente, esto incluye tratamiento de los datos.

Clase Pgsql: Gestiona desde php las funciones de postgresql.

Clase CwSerializable: Se encarga de serializar todas aquellas clases que pueden ser serializadas, con el objetivo de transferir objetos a través de SOAP[11], permitiendo la comunicación entre el Client y el Server del plugin.

Clase Serverlocations: Es la clase servidora, la cual tiene como principal función la conexión con la base de datos para efectuar los cambios requeridos y enviar las respuestas necesarias al Clientlocations.

Clase Clientlocations: Es la encargada de recibir los parámetros enviados a través de las peticiones Ajax realizadas desde el cliente, creando un objeto de tipo Requestlocations el cual envía al Serverlocations.

Clase Requestlocations: Se encarga de transportar los datos recogidos en Clientlocations desde la interfaz a la clase Serverlocations (con propósito general).

Clase Resultlocations: Se encarga de transportar los datos generados en Serverlocations a la clase Clientlocations.

Clase AJAXHelper: Tiene como propósito enviar las respuestas de los plugins “AJAX”, para sustentar a los plugins que responden a las peticiones del usuario.

Clase index.html: Es la encargada de mostrar en el mapa la información necesaria para la gestión de los parques.

AJAXPlugin.Localizar: Es la encargada de gestionar el pedido y las respuestas a las peticiones del usuario por ajax.

2.5.1 Patrones de diseño.

Los patrones de diseño son muy útiles en el proceso de desarrollo de software, dado que ofrecen soluciones probadas y documentadas, ante problemas comunes. Un patrón de diseño identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades. De forma general, se clasifican en dos grupos:

[11] SOAP: Simple Object Access Protocol o Protocolo de Acceso a Objetos Simple.

Patrones de Principios Generales para Asignar Responsabilidad (GRASP) y Patrones de Diseño Pandilla de los Cuatro o Gound-of-Four (GoF) (Larman, 2012).

- **Patrones GRASP empleados:**

Experto: se emplea para asignar una responsabilidad a la clase que posee la información necesaria para cumplir con la responsabilidad (Larman, 2012). Dentro del sistema se utiliza este patrón en clases, que se responsabilizan en proporcionar los datos cargados de una fuente de datos, así como modificar los mismos.

Creador: asigna la responsabilidad de creación a la clase que agrega, contiene o registra el objeto que se va a crear. (Larman, 2012). Dentro del sistema se evidencia su uso en los casos de la instanciación de las clases dentro de la interfaz, los controladores.

Bajo acoplamiento: Se utiliza para asignar las responsabilidades de modo que se mantenga bajo acoplamiento (Larman, 2012). Este patrón se utiliza en la asignación de responsabilidades a clases de manera que un cambio en una de estas genere poco cambio en otras. Se evidencia en la interacción entre las clases con una mínima dependencia entre ellas.

Alta cohesión: Se emplea para asignar las responsabilidades de modo que se mantenga una alta cohesión (Larman, 2012). Se da una alta cohesión funcional cuando los elementos de un componente presentan responsabilidades altamente relacionadas, y no hacen una gran cantidad de trabajo.

- **Patrones GoF empleados:**

Acción: Se utiliza en el proceso de petición de una información cualquiera al sistema por un cliente, mediante la Interfaz Gráfica de Usuario (GUI) (Larman, 2012). Este patrón es utilizado en la clase *AjaxHelper*, la cual se encarga de manejar las peticiones desde la interfaz de usuario y el servidor.

Solitario: Este patrón se aplica para garantizar el acceso único a una clase mediante una única instancia, lo cual permite controlar el acceso a las clases (Reynoso, y otros, 2013). En el presente caso, se utiliza para modificar el *framework* CartoWeb, con el fin de crear el objeto *"mapa"* para que no se cree cada vez que se hace una petición en la aplicación. Esto se evidencia en la clase *ServerContext*.

2.6 Modelo de datos.

El modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí (Roca Díaz., y otros, 2012). Por tal motivo, se le considera un aspecto fundamental para el desarrollo de cualquier aplicación que necesite almacenar datos.

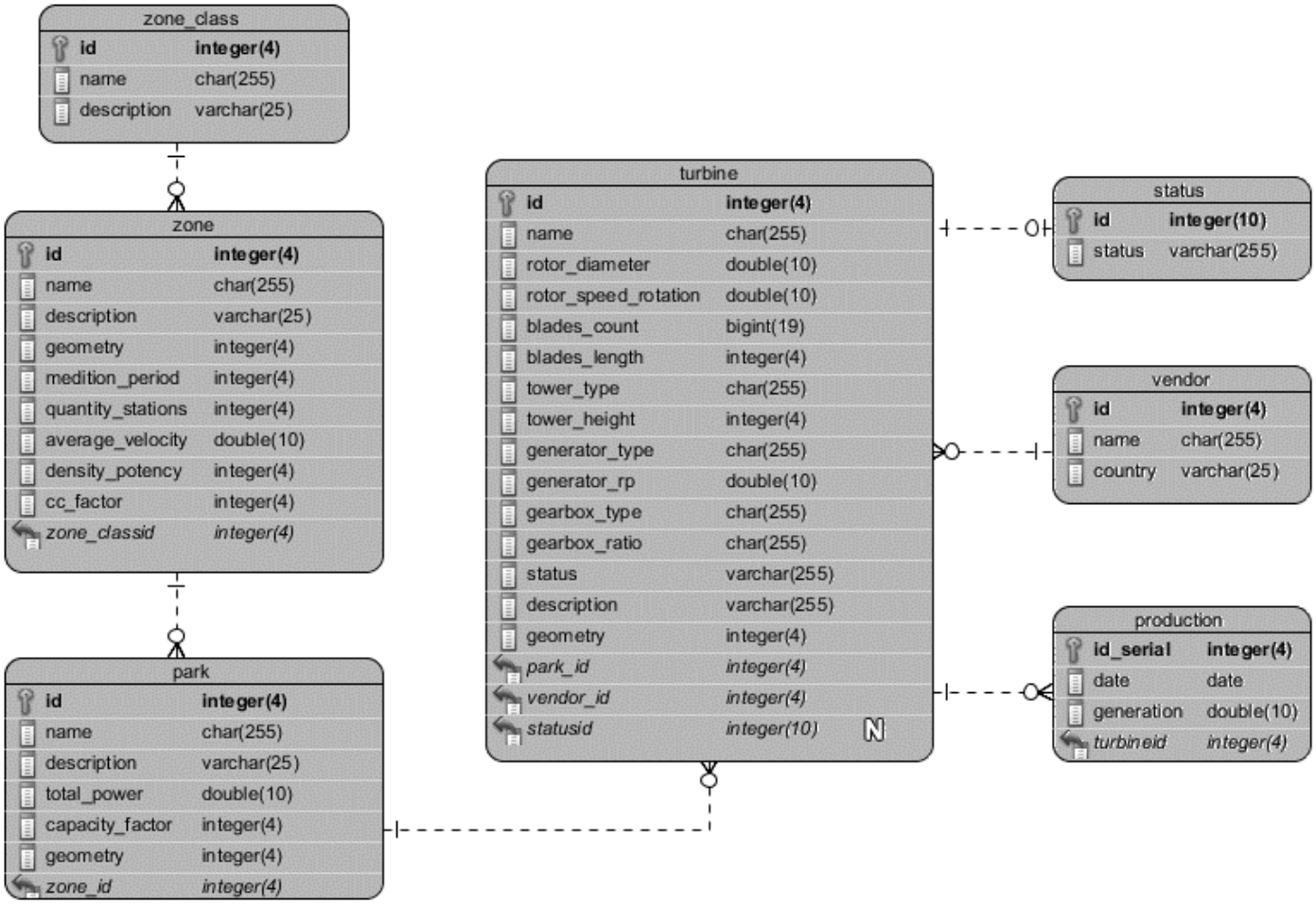


Figura 2: Modelo Entidad-Relación de la base de datos.

2.7 Modelo de despliegue.

"El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo" (Hernández Couce, 2012). Se representa como un conjunto de nodos unidos por conexiones de comunicación que describen las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. A continuación se presenta el diagrama correspondiente al despliegue del sistema.

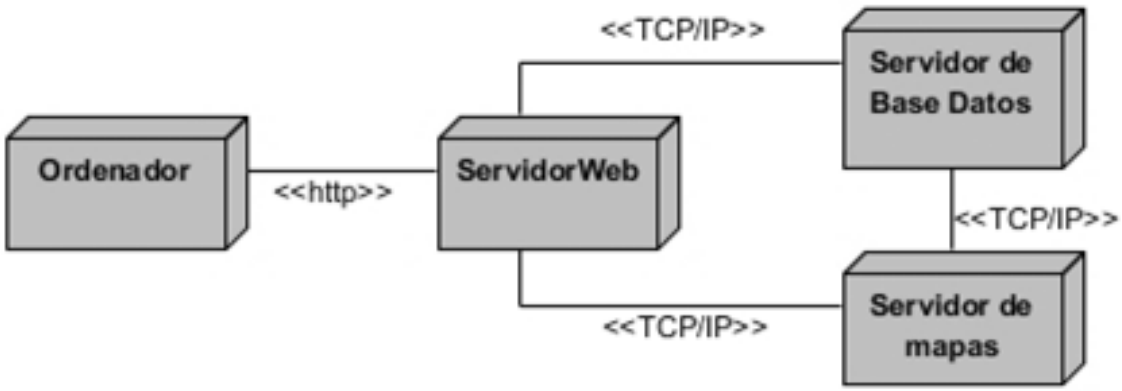


Figura 3: Modelo de despliegue.

2.8 Diagrama de componentes.

El diagrama de componentes muestra la vista física de la aplicación a través de los componentes y sus relaciones; representada como un sistema de software dividido en elementos que muestra las dependencias entre estos. En la figura 4, se muestra el diagrama de componentes correspondiente al sistema:

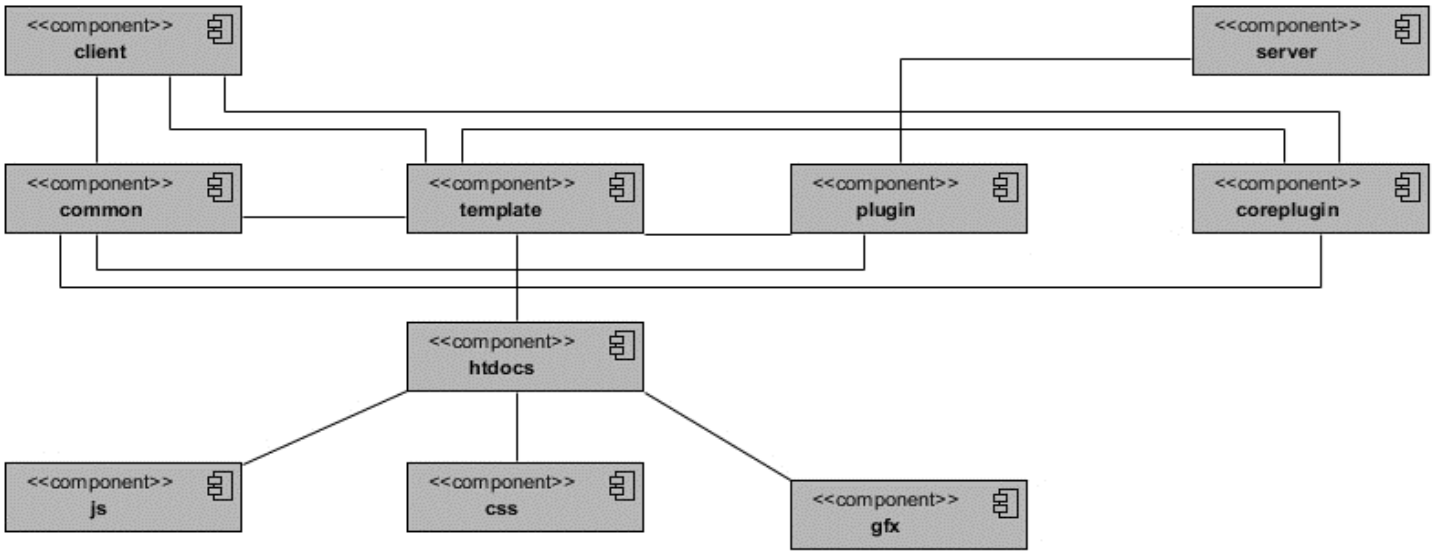


Figura 4 : Modelo de diagrama de componentes.

2.9 Estándar de codificación

Mediante el uso de convenciones de nomenclatura o estándar de codificación, se asegura que el código sea entendible y más fácil de leer; tanto para los propios programadores de un software, como para otros programadores.

La legibilidad del código es importante porque significa que se dedica menos tiempo a intentar averiguar qué hace el código, dejando más tiempo para corregirlo o modificarlo.

CamelCase: es una convención de nomenclatura en la que se forma un nombre de varias palabras, que se unen sin espacios. Según (Leahy, 2016) existen variedades en CamelCase.

- **Lowercase:** es cuando todas las letras de una palabra son escritas sin mayúsculas.
- **Uppercase:** es cuando todas las letras de una palabra son escritas en mayúsculas. Cuando hay más de dos palabras en el nombre se usa el caracter guion bajo para separarlas.
- **CamelCase** (también conocido como **UpperCamelCase**): es donde cada nueva palabra se escribe con letra mayúscula.
- **MixedCase** (también conocido como **lowerCamelCase**): es lo mismo que CamelCase excepto que la primera letra del nombre está en minúscula.

En el sistema se utiliza, UpperCamelCase para el nombre de las clases y lowerCamelCase para el nombre de métodos y variables.

```
1   TurbineView = Ext.extend(Ext.CRUDWindow, {
```

Figura 5: Declaración de una clase empleando UpperCamelCase

```
164     _this.vendorStore = new Ext.data.JsonStore({
```

Figura 6: Declaración de variables empleando lowerCamelCase

2.10 Conclusiones del capítulo.

El presente capítulo concluye con el diseño, modelación y construcción del sistema, derivando en la primera versión del sistema que dará solución al problema planteado. El uso de patrones permitió elaborar los diagramas de clases de diseño fiables para la implementación de la aplicación. Con la selección del estándar de codificación se logra homogeneizar el código de manera legible y fácil de entender para otros desarrolladores. Se obtiene también, la documentación establecida por la metodología, arrojando como resultado un grupo de artefactos relacionados con el desarrollo del sistema.

CAPÍTULO 3. Prueba del sistema desarrollado.

3.1 Introducción.

Las pruebas son un elemento fundamental para garantizar la calidad del software. Están orientadas a descubrir errores en el sistema, pero no aseguran la ausencia de defectos en él. En el presente capítulo se ejecutan las pruebas aplicadas a la solución lograda, se muestran los diseños de caso de prueba correspondientes, los resultados que arrojaron y las no conformidades que generaron.

3.2 Pruebas al sistema.

Las pruebas son aplicadas en diferentes escenarios y niveles, dependiendo del objetivo específico al que se le desee dar cumplimiento. Asimismo, es necesario definir los métodos de prueba que se emplean y las técnicas ligadas a dichos métodos. Al sistema SIG_GIPEC se le aplicaron las pruebas de sistema empleando como método, la prueba de caja negra; y como técnica la Partición de Equivalencia.

- Prueba de sistema

La prueba de sistema verifica el programa final, luego de que todo el software está funcionando como un todo o luego de que subconjuntos bien formados hayan sido implementados. Para aplicar esta prueba al sistema desarrollado, se utilizó el método de caja negra.

- Pruebas de caja negra.

Las pruebas de caja negra se aplican sobre la interfaz del software. Al conocerse la función específica para la que se diseñó el producto, la aplicación de estas pruebas permiten demostrar que cada función de dicho producto es plenamente operacional y a su vez se buscan lo errores.

Las pruebas de caja negra permiten encontrar:

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

Para desarrollarlas existen las técnicas:

- Técnica de la Partición de Equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Se seleccionó la Técnica de Partición de Equivalencia para ser aplicada al SIG_GIPEC porque es una de las más efectivas, debido a que permite examinar los valores válidos e inválidos de las entradas existentes en el software y descubre errores que son detectables ejecutando muchos casos de prueba, de forma inmediata.

3.3 Diseños de Casos de Prueba.

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

Tabla 4: DCP_Gestionar proveedor: SC1_Adicionar proveedor.

Escenario	Descripción	Nombre	País	Respuesta del sistema	Flujo central
EC1.1:Añadir proveedor satisfactoriamente.	El usuario selecciona la opción "Gestionar proveedor", introduce los	V	V	El sistema muestra una ventana con un formulario para añadir un nuevo proveedor.	1.Autenticarse con uno de los roles descritos en las condiciones de ejecución.
		Gamesa	España		

	datos correctamente y presiona el botón ”Aceptar”.			Añade el un nuevo proveedor y muestra un mensaje de confirmación.	2.Seleccionar la funcionalidad ”Gestionar proveedor”. 3.Introducir los datos en el formulario que muestra el sistema. 4.Seleccionar la opción “Adicionar”.
EC1.2: Introducir datos incorrectos.	El usuario introduce datos incorrectos en el formulario y da el botón “Aceptar”.	V Vergnet	I uni”%.&((=)	El sistema señala de color rojo el campo incorrecto con un logo. Al poner el mouse por encima de él muestra un mensaje informando del error y mantiene el botón”Adicionar” inactivo.	1. Autenticarse con uno de los roles descritos en las condiciones de ejecución. 2. Seleccionar la funcionalidad”Gestionar proveedor”. 3.Introducir los datos en el formulario que muestra el sistema. 4.Seleccionar la opción “Adicionar”.

3.4 Resultado de las pruebas

Posterior a la selección del método de caja negra y la técnica de partición de equivalencia, se realizaron 4 iteraciones donde se detectaron 12 errores. A continuación se representan mediante una gráfica las iteraciones mencionadas.

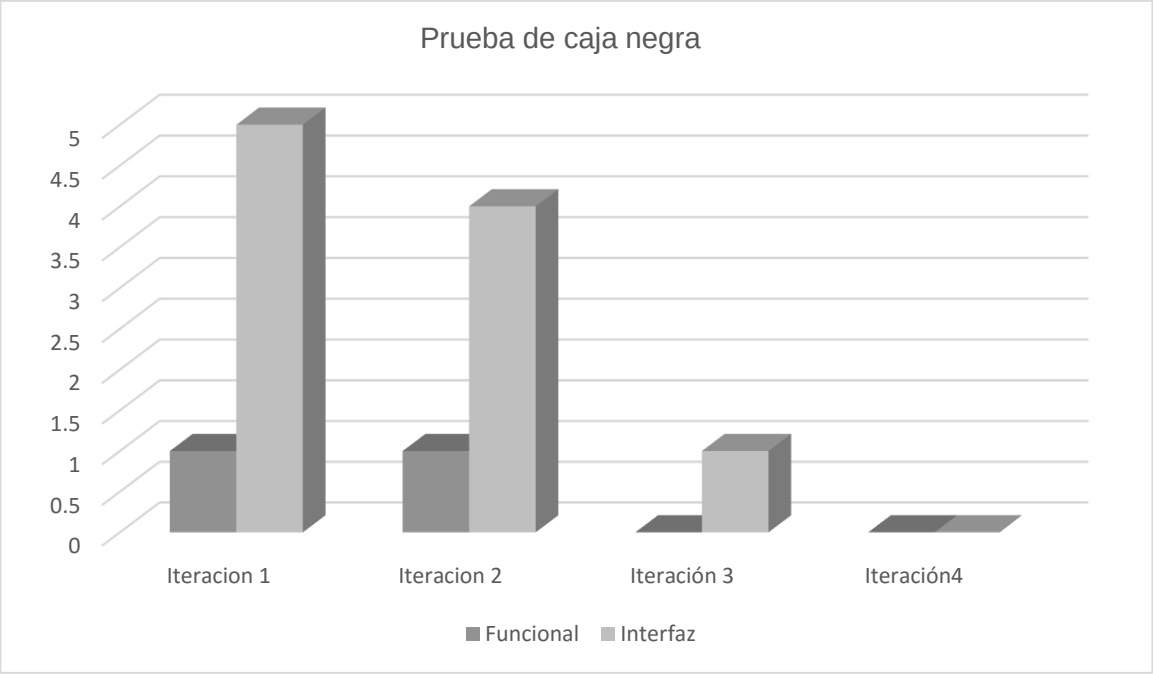


Figura 7: Resultados de la prueba de caja negra.

3.4.1 Conjunto de no conformidades detectadas

Tabla 5: Conjunto de no conformidades detectadas.

Tipo	No	No conformidad	Complejidad			Respuesta
			A	M	B	
Funcional	1	Cuando el usuario selecciona la opción Gestionar Zona y adiciona una nueva zona, después de introducir los datos no se activa el botón adicionar	x			Se corrigió
	2	Cuando el usuario selecciona la opción Gestionar Parques y adiciona el parque, después de introducir los datos se activa el botón *Modificar* y *Adicionar*, cuando solo se debe activar el último.	x			Se corrigió
Interfaz	3	Cuando el usuario selecciona la opción Gestionar Zona, después de capturar la geometría, no se expande la ventana para adicionar.		x		Se corrigió

	4	Cuando el usuario selecciona la opción Gestionar Zonas, puede apreciar 4 faltas de ortografía en los nombres de los campos Período de Medición, Velocidad Promedio, Clasificación, Densidad de potencia.			x	Se corrigió
	5	Cuando el usuario selecciona la opción Gestionar turbinas, puede apreciar 3 faltas de ortografía en los nombres de los campos Diámetro del rotor, Velocidad de rotación del rotor y Potencia nominal del generador.			x	Se corrigió
	6	Cuando el usuario selecciona la opción Gestionar Proveedor, puede apreciar 1 falta de ortografía en el nombre del campo País.			x	Se corrigió

3.5 Conclusiones parciales.

La etapa de pruebas permitió verificar el resultado de la implementación, proporcionando una perspectiva de cómo se realizó dicha implementación. Guiaron, además, los pasos para la revisión de los elementos del software. El reconocimiento de las principales fallas cometidas en dicho proceso y su posterior resolución, posibilito que las funcionalidades del sistema respondan a los objetivos planteados inicialmente.

CONCLUSIONES GENERALES

Al término de esta investigación se considera que fueron alcanzados los objetivos trazados al desarrollar el Sistema de Información Geográfica para la gestión de información de los parques eólicos en Cuba. El proceso estuvo guiado por la ejecución de un grupo de aspectos, detallados a continuación:

3 El estudio de los fundamentos teóricos relacionados con el tema a tratar, permitió la selección de una metodología, de herramientas y tecnologías adecuadas, claves para la creación del sistema.

4 Mediante el análisis y diseño de la propuesta de solución se obtuvo una visión general del sistema sentando las bases para su posterior implementación. El diseño propuesto es claro y simple.

5 La identificación de los requisitos del sistema permitió generar los artefactos ingenieriles correspondientes a la metodología utilizada, los cuales posibilitaron organizar y entender los principales conceptos que se manejan en el sistema y sus relaciones.

6 Los diferentes artefactos generados durante el proceso de desarrollo pueden ser empleados para la comprensión, modificación o la incorporación de nuevas funcionalidades.

7 El proceso de pruebas desarrollado permitió identificar y resolver los errores en la implementación aumentando la calidad del producto final obtenido.

8 Se ha obtenido como producto final un Sistema de Información Geográfica para la gestión de información de los parques eólicos en Cuba, que cumple con las necesidades identificadas en INEL.

RECOMENDACIONES

Una vez concluida la investigación y basándose en las experiencias acumuladas a lo largo del desarrollo de la misma, se propone la siguiente recomendación:

Como proyección futura, para mejorar el sistema se prevé la interconexión entre el SIG_GIPEC y los sistemas SCADA´s a través del protocolo TCP/IP, para eliminar la entrada de forma manual de los datos de la generación eléctrica diaria de los parques eólicos por el operador y evitar así cualquier error humano. De esta forma se automatiza y agiliza el trabajo.

REFERENCIAS BIBLIOGRÁFICAS

Apache. 2013. Documentación del Servidor Apache. [En línea] 2013. [Citado el: 2017 de abril de 23.] http://httpd.apache.org/docs/2.0/es/.

Bass, L., P. Clements y Kazaman, R. 2015. *Software Architecture in Practice.* s.l. : Adison - Wesly, 2015.

Biblioteca virtual- BR. 2014. Banco de la República(Cartografía). *Banco de la República(Cartografía).* [En línea] 2 de septiembre de 2014. [Citado el: 5 de mayo de 2017.] http://www.banrepcultural.org/blaavirtual/ayudadetareas/geografia/cartografia.

Castañeda Rodríguez, Siomelis. 2014. *Sistema de Información Geográfica para Campismo Popular.* La Habana : s.n., 2014.

Eólica, Asociacion Empresarial. 2012. [En línea] 3 de 8 de 2012. [Citado el: 12 de octubre de 2016.] http://www.laregion.es/noticia/64677/parque/e%C3%B3lico/. 4.

Figueredo, Conrado Moreno. 2013. Cubasolar. [En línea] 2013. [Citado el: 13 de diciembre de 2016.] http://www.cubasolar.cu/biblioteca/Energia/Energia38/HTML/articulo04.htm.

Fuentenebro, Laura de la Cruz. 2014. *DISEÑO Y MODELACION DE LA POTENCIA.* Madrid : s.n., 2014.

Gámez, Maria Rodriguez. 2012. *LOS SIG Y LA INTEGRACIÓN IBEROAMERICANA A PARTIR DEL USO DE LAS FUENTES RENOVABLES DE ENERGÍA.* Madrid : CIEMAT, 2012. 3.

González, Rolando Alfredo Hernández León y Sayda Coello. 2012. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA.* La Habana : Universitaria, 2012. ISBN 978-959-16-1307-3. -- 110 pág..

Hernández Barrios, Misbel. 2012. *Desarrollo de un Sistema de Información.* Ciudad de La Habana : s.n., 2012.

Hernández Couce, Daryl. 2012. *Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas sobre dispositivos móviles basado en los servicios Web estandarizados por la Open Geospatial Consortium.* [Documento] La Habana : s.n., 2012.

I. Jacobson, G. Bosh. 2011. *El proceso Unificado de Desarrollo de Software.* Madrid : PEARSON EDUCACION.SA, 2011. 458 p. 84-7829-036-2.

IBM. 2013. IBM. [En línea] 2013. [Citado el: 17 de Noviembre de 2016.] http://www-01.ibm.com/software/rational/rup/.

Jacobson, Ivar, Booch, Grady y Rumbaugh, Jame. 2012. *El Proceso Unificado de Desarrollo. Vol I, Vol II.* La Habana : Felix Varela, 2012.

Jaime M. Tan Nozawa. 2007. PHP Experto. [En línea] 5 de noviembre de 2007. [Citado el: 14 de mayo de 2017.] http://phpexperto.blogspot.com/2007/11/php-mapscript-de-mapserver-parte-1.html.

La Arquitectura de Software. 2014. La Arquitectura de Software. [En línea] 2014. [Citado el: 5 de Marzo de 2017.] arquitecturadesoftware.com.

Larman, Craig. 2012. *UML y Patrones. 2da Edición.* España : Pearson Educación, 2012. 28.

Leahy, Paul. 2016. Using Java Naming Conventions. [En línea] 25 de julio de 2016. [Citado el: 9 de junio de 2017.] https://www.thoughtco.com/using-java-naming-conventions-2034199.

MappingGIS. 2015. MappingGIS. [En línea] 2 de octubre de 2015. [Citado el: 14 de mayo de 2017.] http://mappinggis.com/2012/09/por-que-utilizar-postgis/.

Matemáticas, Departamento de Geofísica Facultad de Ciencias Físicas y Matemáticas Facultad de Ciencias Físicas y. 2012. El Explorador Eólico 2012. Chile : s.n., 2012.

Miranda, Ubaldo. 2012. El Sistema de Información Geográfica para las Energías Renovables (SIGER) en México. México : Boletín IIE, 2012.

Mozilla Developer Network. 2005. MDN. [En línea] 2005. [Citado el: 11 de Noviembre de 2016.] https://developer.mozilla.org/es/docs/JavaScript.

Olaya, Víctor. 2011. *Sistemas de Información Geográfica.* [Documento] 2011.

Oracle Corporation. 2017. NetBEANS. *NetBEANS.* [En línea] Oracle, 2017. [Citado el: 1 de junio de 2017.] file:///C:/Users/Toshiba/Desktop/NetBeans%20IDE%20-%20PHP%20Development.html.

PostgreSQL. 2009. PostgreSQL-es. [En línea] 2009. [Citado el: 7 de Diciembre de 2016.] http://www.postgresql.org.es/sobre_postgresql.

Pressman, Roger S. 2010. *Ingeniería de Software, un enfoque práctico.* 2010.

Reynoso, Carlos y Kiccillof, Nicolás. 2013. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft Versión.* [Documento] Buenos Aires : Universidad de Buenos Aires, 2013.

Roca Díaz., Yanet y Pérez González, Miguel Yoandy. 2012. *Sistema de Información Geográfica para el transporte obrero de la Universidad de las Ciencia Informáticas. Versión 2.0.* [Documento] La Habana : s.n., 2012.

Sánches, Tamara Rodríguez. 2015. *:Metodología de desarrollo para la Actividad productiva UCI. :Metodología de desarrollo para la Actividad productiva UCI. :Metodología de desarrollo paMetodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.

Scribd inc. 2017. SCRIBD. *SCRIBD.* [En línea] 2017. [Citado el: 5 de junio de 2017.] https://es.scribd.com/document/324351073/ue-es-la-Georeferenciacion-docx.

Sphinx. 2014. MapServer. [En línea] 2014. [Citado el: 17 de Enero de 2017.] http://www.mapserver.org/es/about.html#about.

The free dicctionary. 2014. The free dicctionary. [En línea] 2014. [Citado el: 24 de Abril de 2017.] http://thefreedictionary.com/zoom.

The PHP Group. 2017. *PHP.* 2017.

VisualParadigm. 1999. VisualParadigm. [En línea] 1999. [Citado el: 5 de Febrero de 2017.] http://www.visual-paradigm.com/product/vpuml/.

BIBLIOGRAFÍA

.NET Design Patterns . [En línea] [Citado el: 2017 de junio de 9.] http://www.dofactory.com/net/design-patterns.

Apache. 2013. Documentación del Servidor Apache. [En línea] 2013. [Citado el: 2017 de abril de 23.] http://httpd.apache.org/docs/2.0/es/.

Bass, L., P. Clements y Kazaman, R. 2015. *Software Architecture in Practice.* s.l. : Adison - Wesly, 2015.

Biblioteca virtual- BR. 2014. Banco de la República(Cartografía). *Banco de la República(Cartografía).* [En línea] 2 de septiembre de 2014. [Citado el: 5 de mayo de 2017.] http://www.banrepcultural.org/blaavirtual/ayudadetareas/geografia/cartografia.

Bravo, Javier Dominguez. 2012. *Los Sistemas de Información Geográfica y las Energías Renovables.* Madrid : Ciemat, 2012.

Castañeda Rodríguez, Siomelis. 2014. *Sistema de Información Geográfica para Campismo Popular.* La Habana : s.n., 2014.

ecotecnia | Technology For Life. [En línea] [Citado el: 2017 de junio de 12.] https://technology4life.wordpress.com/tag/ecotecnia/.

Eólica, Asociacion Empresarial. 2012. [En línea] 3 de 8 de 2012. [Citado el: 12 de octubre de 2016.] http://www.laregion.es/noticia/64677/parque/e%C3%B3lico/. 4.

Ferris, David. 2015. ELECTRICITY. *Cuba wants clean energy. Can the U.S. deliver?* [En línea] 25 de junio de 2015. [Citado el: 2017 de JUNIO de 15.] https://www.eenews.net/stories/1060020853.

Figueredo, Conrado Moreno. 2013. Cubasolar. [En línea] 2013. [Citado el: 13 de diciembre de 2016.] http://www.cubasolar.cu/biblioteca/Energia/Energia38/HTML/articulo04.htm.

Fuentenebro, Laura de la Cruz. 2014. *DISEÑO Y MODELACION DE LA POTENCIA.* Madrid : s.n., 2014.

Galeano Gil, Germán , Díaz Márquez, Pablo y Sánchez Alonso, José Carlos. 2016. *Manual de HTML4.* Barcelona : Anaya Multimedia-Anaya Interactiva, 2016. 1ª ed.

Gamesa / Products and services / Wind turbines. [En línea] [Citado el: 2017 de junio de 11.] http://www.gamesacorp.com/en/products-and-services/wind-turbines/products-and-services-wind-turbines-catalogue.html.

Gámez, Maria Rodriguez. 2012. *LOS SIG Y LA INTEGRACIÓN IBEROAMERICANA A PARTIR DEL USO DE LAS FUENTES RENOVABLES DE ENERGÍA.* Madrid : CIEMAT, 2012. 3.

GofPatterns | Gof Patterns | Gang of Four Patterns | Design Patterns. [En línea] [Citado el: 2016 de junio de 15.] http://www.gofpatterns.com/.

González, Rolando Alfredo Hernández León y Sayda Coello. 2012. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA.* La Habana : Universitaria, 2012. ISBN 978-959-16-1307-3. -- 110 pág..

2017. Havana Energy. *Wind Farm – Havana Energy.* [En línea] 2017. http://havana-energy.com/wind-farm/.

Hernández Barrios, Misbel. 2012. *Desarrollo de un Sistema de Información.* Ciudad de La Habana : s.n., 2012.

Hernández Couce, Daryl. 2012. *Sistema de Información Geográfica para la Universidad de las Ciencias Informáticas sobre dispositivos móviles basado en los servicios Web estandarizados por la Open Geospatial Consortium.* [Documento] La Habana : s.n., 2012.

I. Jacobson, G. Bosh. 2011. *El proceso Unificado de Desarrollo de Software.* Madrid : PEARSON EDUCACION.SA, 2011. 458 p. 84-7829-036-2.

IBM. 2013. IBM. [En línea] 2013. [Citado el: 17 de Noviembre de 2016.] http://www-01.ibm.com/software/rational/rup/.

Jacobson, Ivar, Booch, Grady y Rumbaugh, Jame. 2012. *El Proceso Unificado de Desarrollo. Vol I, Vol II.* La Habana : Felix Varela, 2012.

Jaime M. Tan Nozawa. 2007. PHP Experto. [En línea] 5 de noviembre de 2007. [Citado el: 14 de mayo de 2017.] http://phpexperto.blogspot.com/2007/11/php-mapscript-de-mapserver-parte-1.html.

Jose Luis Almazán, Carmen y Caba Palomin, y Hilda Araceli Garrete. 2012. *Sistema de Información Geográfica en la Gestión Integral del Litoral.* Madrid : s.n., 2012.

La Arquitectura de Software. 2014. La Arquitectura de Software. [En línea] 2014. [Citado el: 5 de Marzo de 2017.] arquitecturadesoftware.com.

Larman, Craig. 2012. *UML y Patrones. 2da Edición.* España : Pearson Educación, 2012. 28.

Leahy, Paul. 2016. Using Java Naming Conventions. [En línea] 25 de julio de 2016. [Citado el: 9 de junio de 2017.] https://www.thoughtco.com/using-java-naming-conventions-2034199.

López, Gerardo Sarabia. 2011. *Búsqueda y ponderación de información contenida en Bases de datos Espaciales,.* México : Centro de Investigación en computación, 2011. Tesis.

Luis. 2014. Wind Power Monthly. *Analysis: Cuba looks beyond US for wind alliances.* [En línea] 1 de diciembre de 2014. [Citado el: 2017 de junio de 10.] http://www.windpowermonthly.com/article/1324531.

Madruga, Ramon Pichs. 2012. Cubasolar. [En línea] 2012. [Citado el: 12 de octubre de 2016.] 5www.cubasolar.cu/biblioteca/Ecosolar/Ecosolar39_HTML_articul.pdf. 1.

2016. Manual de HTML. [En línea] 24 de marzo de 2016. [Citado el: 2017 de abril de 23.] http://www.desarrolloweb.com/manuales/21/.

MappingGIS. 2015. MappingGIS. [En línea] 2 de octubre de 2015. [Citado el: 14 de mayo de 2017.] http://mappinggis.com/2012/09/por-que-utilizar-postgis/.

Matemáticas, Departamento de Geofísica Facultad de Ciencias Físicas y Matemáticas Facultad de Ciencias Físicas y. 2012. El Explorador Eólico 2012. Chile : s.n., 2012.

Miranda, Ubaldo. 2012. El Sistema de Información Geográfica para las Energías Renovables (SIGER) en México. México : Boletín IIE, 2012.

Moreta Fernández, Maricet . 2013. *Desarrollo de una aplicación web que permita incrementar la celeridad en el procesamiento de la información generada en las secciones sindicales de la Universidad de las Ciencias Informáticas.* La Habana : UCI, 2013. 1.

Mozilla Developer Network. 2005. MDN. [En línea] 2005. [Citado el: 11 de Noviembre de 2016.] https://developer.mozilla.org/es/docs/JavaScript.

Olaya, Víctor. 2011. *Sistemas de Información Geográfica.* [Documento] 2011.

Oracle Corporation. 2017. NetBEANS. *NetBEANS.* [En línea] Oracle, 2017. [Citado el: 1 de junio de 2017.] file:///C:/Users/Toshiba/Desktop/NetBeans%20IDE%20-%20PHP%20Development.html.

Perez, Humberto Oyazo. 2012. *Modelización matemática para evaluar energías para sistemas eólicos e híbridos eólico-diesel.* Madrid : s.n., 2012. 2.

php. 2013. php. [En línea] 2013. [Citado el: 8 de Noviembre de 2016.] http://php.net/manual/es/intro-whatis.php.

PostgreSQL. 2009. PostgreSQL-es. [En línea] 2009. [Citado el: 7 de Diciembre de 2016.] http://www.postgresql.org.es/sobre_postgresql.

Pressman, Roger S. 2010. *Ingeniería de Software, un enfoque práctico.* 2010.

Real Academia Española. 2013. Real Academia Española5. [En línea] 2013. http://rae.es/recursos/diccionarios/drae. 28014.

REC. 2014. Renewable Energy Caribbean. *Las Tunas Wind Farm, Cuba .* [En línea] 6 de octubre de 2014. [Citado el: 2017 de junio de 15.] https://renewableenergycaribbean.com/2014/10/06/las-tunas-wind-farm-cuba/.

2016. Report, Cuba Business. *Renewable Energy in Cuba - Wind Farms and Solar Power.* [En línea] 8 de junio de 2016. [Citado el: 15 de junio de 2017.] http://www.cubabusinessreport.com/cubas-green-energy-strategy-to-2030/.

Reynoso, Carlos y Kiccillof, Nicolás. 2013. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft Versión.* [Documento] Buenos Aires : Universidad de Buenos Aires, 2013.

Roca Díaz., Yanet y Pérez González, Miguel Yoandy. 2012. *Sistema de Información Geográfica para el transporte obrero de la Universidad de las Ciencia Informáticas. Versión 2.0.* [Documento] La Habana : s.n., 2012.

Sánches, Tamara Rodríguez. 2015. *:Metodología de desarrollo para la Actividad productiva UCI. :Metodología de desarrollo para la Actividad productiva UCI. :Metodología de desarrollo paMetodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.

Scribd inc. 2017. SCRIBD. *SCRIBD.* [En línea] 2017. [Citado el: 5 de junio de 2017.] https://es.scribd.com/document/324351073/ue-es-la-Georeferenciacion-docx.

Sphinx. 2014. MapServer. [En línea] 2014. [Citado el: 17 de Enero de 2017.] http://www.mapserver.org/es/about.html#about.

The free dicctionary. 2014. The free dicctionary. [En línea] 2014. [Citado el: 24 de Abril de 2017.] http://thefreedictionary.com/zoom.

The PHP Group. 2017. *PHP.* 2017.

2016. The Wind Power. [En línea] 14 de enero de 2016. [Citado el: 2016 de junio de 15 .] http://www.thewindpower.net/windfarm_en_4081_gibara.php.

UCI. 2010. *Registro en el Centro Nacional de Derecho de Autor de Cuba (CENDA).* La Habana : s.n., 2010. DSL-C-42-2010.

Varen Caballero, Eliani. 2010. *Estrategia para la implementación de Sistemas de Información Geográfica del petróleo sobre la base de la Plataforma GeneSIG. TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.* La Habana : s.n., 2010.

Vázquez Manso, Ruben. 2011. *Desarrollo de un Sistema de Información Geográfica para la.* Ciudad de La Habana : s.n., 2011.

VisualParadigm. 1999. VisualParadigm. [En línea] 1999. [Citado el: 5 de Febrero de 2017.] http://www.visual-paradigm.com/product/vpuml/.

Wind Energy The Facts. *SCADA and instruments.* [En línea] [Citado el: 2017 de junio de 15 .] https://www.wind-energy-the-facts.org/scada-and-instruments.html.

2012. Windfarm construction. *Windfarm SCADA system characteristics.* [En línea] 29 de mayo de 2012. [Citado el: 2016 de junio de 15.] http://www.windfarmbop.com/windfarm-scada-system-characteristics/.

Glosario de Términos

Paradigma energético sostenible: es un nuevo modelo para producir y consumir energía donde las energías renovables son las protagonistas.

UI: es un medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

API: Interfaz de Programación de Aplicaciones (IPA) o API (del inglés Application Programming Interface), es el conjunto de funciones que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

OpenSource: software de código abierto que permite a los usuarios utilizar, cambiar y redistribuir el software para cualquier propósito, ya sea en su forma modificada o en su forma original.

GDAL: Geoespacial Data Abstraction Library, es una biblioteca de software para la lectura y escritura de formatos de datos geoespaciales.

OGR: Simple Features Library o Biblioteca de Funciones Sencillas, es una biblioteca de acceso a datos espaciales empleada junto al GDAL, usado por muchas aplicaciones.

OGC: Open Geospatial Consortium, es la definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica.

W3C: estándar que desarrolla especificaciones técnicas y directrices a través de un proceso que ha sido diseñado para maximizar el consenso sobre el contenido de un informe técnico, de forma que se pueda asegurar la alta calidad técnica y editorial, así como obtener un mayor apoyo desde el W3C y desde la comunidad en general.

Plugin: aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

SOAP: paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir.

SCADA´s: Sistemas de Control Supervisorio y Adquisición de Datos son conocidos por el término SCADA, que proviene de las siglas en inglés "Supervisory Control And Data Acquisition". Es un software de

aplicación, diseñado especialmente para ejecutarse sobre ordenadores destinados al control de la producción, proporcionando comunicación con los dispositivos de campo (controladores digitales autónomos, autómatas programables, instrumentación inteligente, etcétera) y controlando el proceso de forma automática desde la pantalla del ordenador.

Datos raster: son modelos de datos espaciales que definen el espacio como un conjunto de celdas del mismo tamaño, ordenadas en filas y columnas. Pueden ser fotografías aéreas digitales, imágenes de satélite, imágenes digitales o incluso mapas escaneados.

Datos vectoriales: son estructuras de datos utilizados para almacenar información geográfica.

Geoportal: es un sitio web cuya finalidad es ofrecer a los usuarios el acceso a una serie de recursos y servicios basados en la información geográfica.

ANEXOS

Anexo 1: **Entrevista sobre problemas existentes en INEL.**

Preguntas:

- ¿Cuál es la misión del departamento de Recursos Renovables?
- ¿Qué problemas presenta el departamento?
- ¿Con que información cuenta hoy el departamento?
- ¿Cómo se obtiene la información de los parques eólicos?
- ¿Poseen algún sistema informático que permita el manejo de esta información?
- ¿Qué objetivo tiene el sistema a desarrollar?
- ¿A qué usuarios va dirigido el sistema?
- ¿Poseen algún sistema semejante al sistema que desean?

Anexo 2: Imágenes del sistema

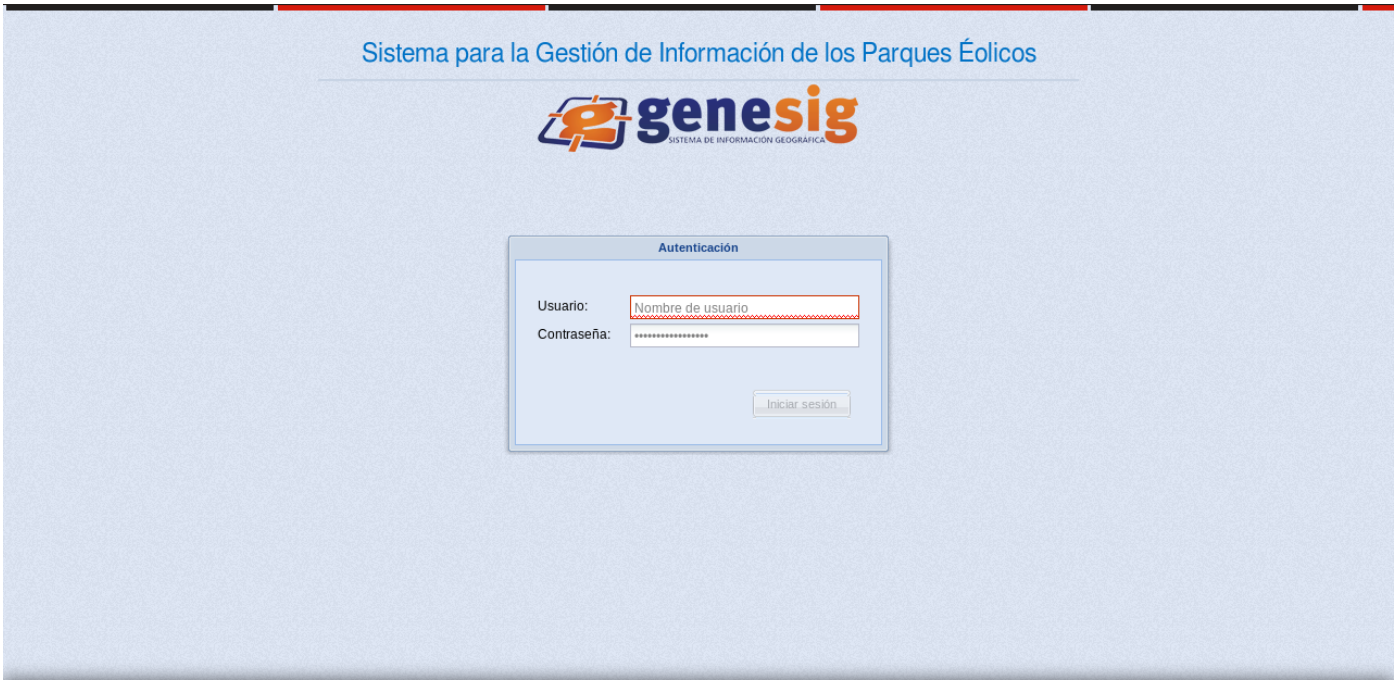


Figura 8: Inicio de sesión

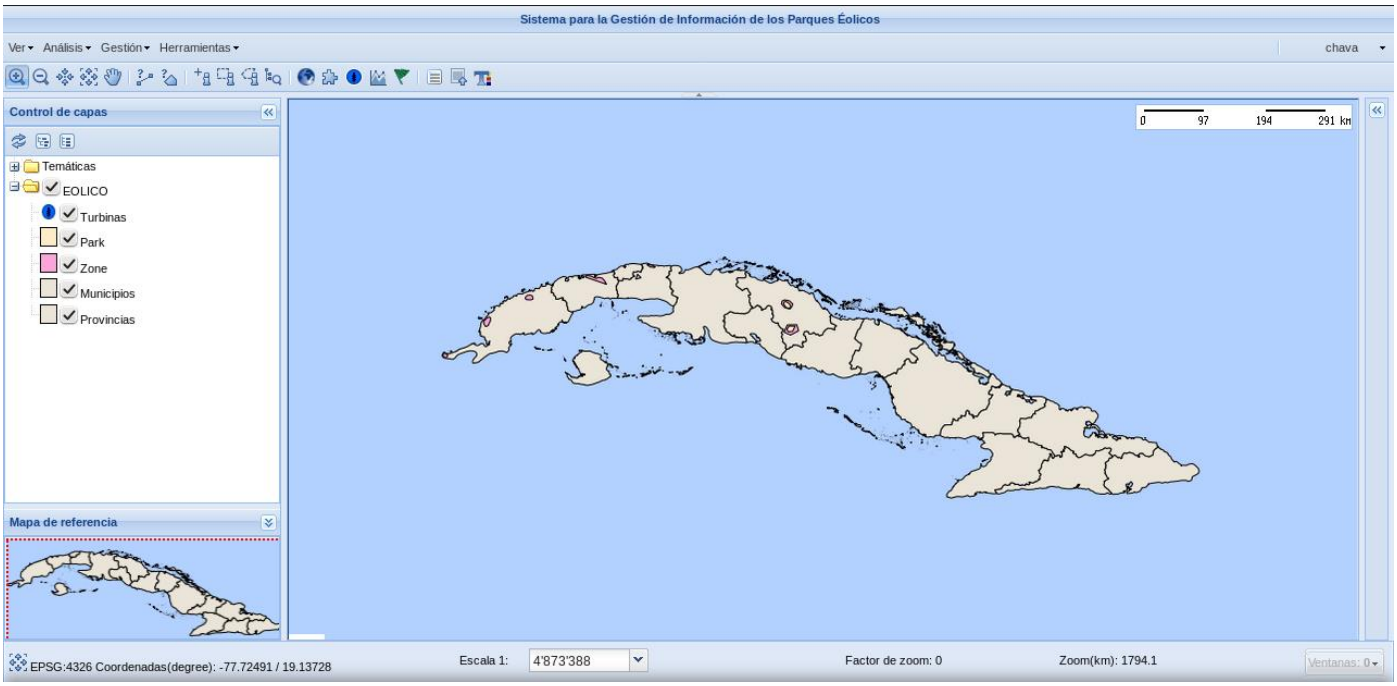


Figura 9: Sistema de Información Geográfica para la gestión de la información de los parques eólico (SIG_GIPEC).

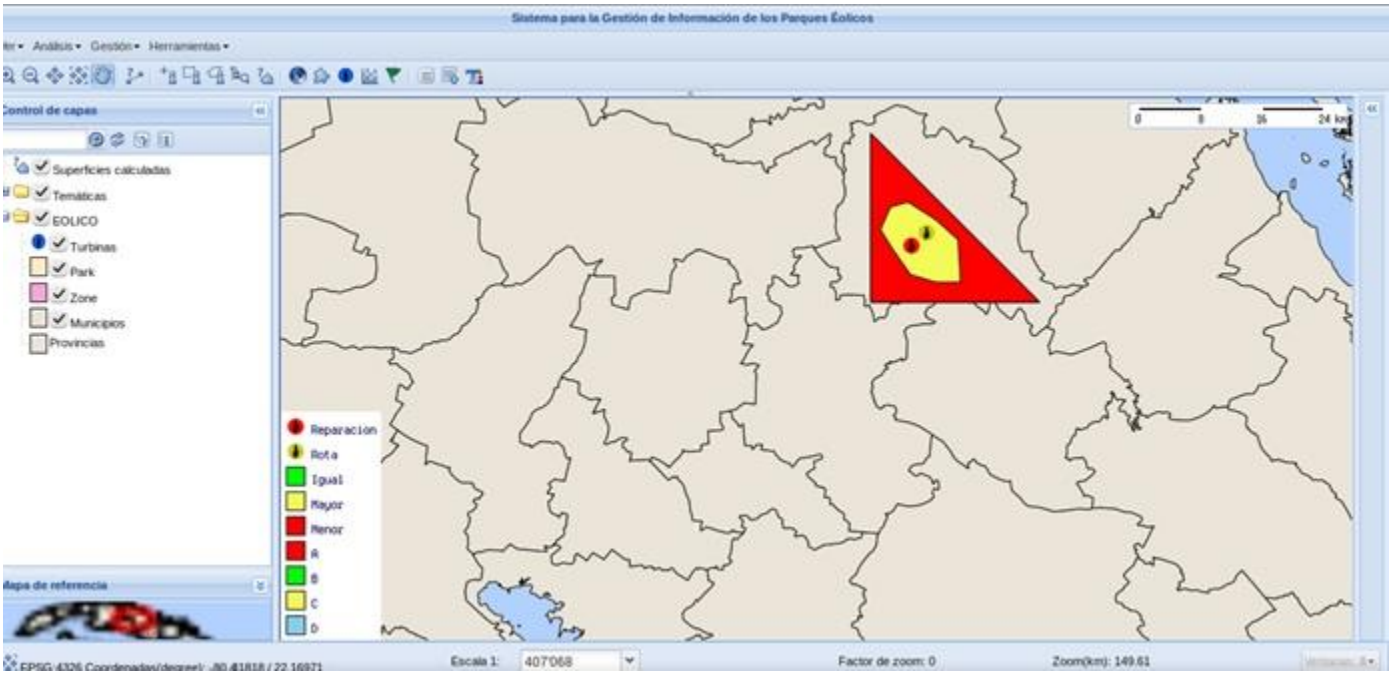


Figura 10: Tematización por colores.

Figura 11: Gestionar proveedor

Figura 12: Cálculo de área y perímetro