

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
Facultad 6



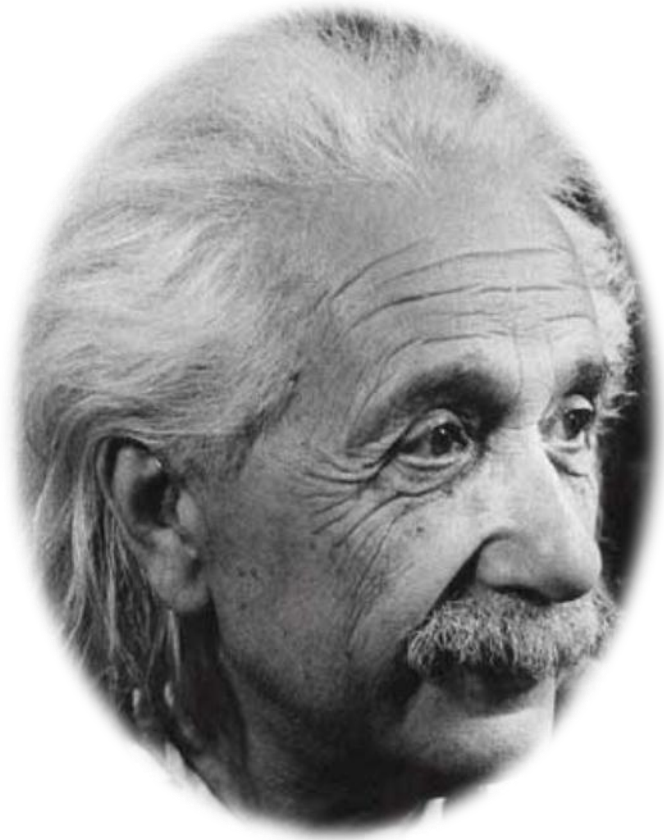
Título: Servicio Web para la búsqueda de objetivos  
geográficos en bases de datos espaciales.

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

**AUTOR:** Alain Vila Acosta

**TUTOR:** Ing. Yoenis Pantoja Zaldívar

**La Habana 19 de junio de 2012  
"Año 54 de la Revolución"**



*La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos.*

*Albert Einstein*

## **Declaración de Autoría**

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Geoinformática de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Alain Vila Acosta

\_\_\_\_\_

Ing. Yoenis Pantoja Zaldívar

\_\_\_\_\_

## **Dedicatoria**

*A mi familia por creer en mí,  
a mi novia por estar siempre a mi lado y  
a mis amigos por apoyarme.*

## **Agradecimientos**

*A mi mamá y mi papá por apoyarme siempre en mis decisiones y confiar siempre en mí, a mis abuelos, dos personas especiales que siempre han estado a mi lado, a mi hermanita por su carisma e incondicionalidad, a mi novia por apoyarme siempre y confiar en mí, además de haberme ayudado en la redacción de este trabajo, a todos los buenos amigos que he encontrado en estos cinco años de universidad, a mis compañeros de aula que aportaron sus ideas en la realización de esta tesis, a los profes del proyecto que siempre estuvieron dispuestos a ayudar en lo que hiciera falta, al profe Alain por su ayuda y apoyo, a mi tutor Yoenis por estar siempre dispuesto a ayudar aun con falta de tiempo.*

*En fin a todos los que han creído en mí y me han ayudado de alguna forma u otra en la realización de este trabajo.*

## **Datos de Contacto**

Tutor: Ing. Yoenis Pantoja Zaldívar E-mail: [ypantojaz@uci.cu](mailto:ypantojaz@uci.cu)

Ingeniero en Informática, Universidad de Holguín, 2006.

Jefe del Departamento de Técnicas de Programación.

Facultad 6. Universidad de las Ciencias Informáticas.

## **Opinión del Tutor**

## **Resumen**

Los SIG o Sistemas de Información Geográfica, según sus siglas, muestran un gran número de funcionalidades que facilitan el trabajo con la información geográfica. Una de las funcionalidades más provechosas que proveen los SIG son los sistemas de búsqueda que incorporan. Las formas que implementan sus resultados en dependencia de la cantidad de información que almacenan, dígase cartografías con un nivel de complejidad alto en cuanto a datos, exigen a los desarrolladores crear soluciones que permitan un mejor y más rápido acceso a la información con la que requieran trabajar. El presente trabajo tiene como objetivo la realización de un servicio Web que sirva para la búsqueda de objetivos geográficos en base de datos espaciales. Dicho trabajo se dividió en cuatro capítulos en los cuales se recogen todas las fases por la que pasó el desarrollo del sistema, desde la investigación teórica hasta la propia construcción de la solución. Se utilizaron las tecnologías pertinentes definidas por los estándares vigentes en la Universidad de las Ciencias Informáticas (UCI) dando al traste con un servicio Web capaz de integrarse a cualquier sistema que necesite realizar búsquedas en bases de datos de cualquier temática, haciendo énfasis en los Sistemas de Información Geográfica.



## Summary

GIS or Geographic Information Systems, according as its acronym, they represent a large number of features that make easier the work with geographic information. Between the capabilities that they possess, we can find the search and display of geographic targets. The search methods of data intended to make of the GISs more functional and manipulated systems, because as they have a lot of information stored, must not wander in search of it so it must find solutions to a better and faster access to information. This investigation have as objective, the making of a Web service that serves to the search of geographic objectives in a spatial database. The work is divided into four chapters which cover all phases of the development happened on the system, from theoretical research until the construction of the actual solution. Relevant technologies were used as defined by current standards at the University of Informatics Sciences (UCI), having as result a web service that can be integrated into any system that needs to search databases of any subject, making emphasis in Geographic Information Systems.

## Índice

|   |           |
|---|-----------|
| <b>CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN .....</b>   | <b>5</b>  |
| 1.1 INTRODUCCIÓN.....   | 5         |
| 1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA .....   | 5         |
| 1.3 CONCEPTOS ASOCIADOS A LA SOLUCIÓN .....   | 6         |
| 1.4 OBJETO DE ESTUDIO .....   | 7         |
| 1.4.1 Procesos asociados a las funcionalidades de búsqueda sobre bases de datos espaciales  | 7         |
| 1.4.2 Servicios Web.....  | 10        |
| 1.5 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES .....   | 12        |
| Métodos de búsqueda de Google Earth .....   | 12        |
| Métodos de búsqueda de la IDE-CHG .....   | 13        |
| Métodos de búsqueda de QGIS .....   | 14        |
| Módulo de búsqueda de la Plataforma de Servicios de Mapas de la UCI.....  | 15        |
| Métodos de búsqueda de SIGUCI.....  | 15        |
| IDERC.....  | 15        |
| 1.6 CONCLUSIÓN DEL ESTUDIO DE LAS SOLUCIONES .....  | 16        |
| 1.7 CONCLUSIONES PARCIALES .....  | 17        |
| <b>CAPÍTULO 2: ANÁLISIS DE LAS TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR.....</b>   | <b>18</b> |
| 2.1 INTRODUCCIÓN .....  | 18        |
| 2.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....  | 18        |
| 2.2.1 Metodologías de desarrollo Ágiles .....   | 18        |
| 2.2.2 Programación Extrema .....  | 20        |
| 2.3 MARCO DE TRABAJO.....   | 22        |
| 2.4 MARCO DE TRABAJO PARA SERVICIOS WEB .....   | 23        |
| 2.5 LENGUAJE DE PROGRAMACIÓN .....  | 24        |
| 2.6 ENTORNO DE DESARROLLO .....   | 24        |
| 2.7 SISTEMA GESTOR DE BASES DE DATOS (SGBD).....  | 25        |
| 2.8 CONCLUSIONES PARCIALES .....  | 25        |
| <b>CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>  | <b>26</b> |
| 3.1 DESCRIPCIÓN DE LA SOLUCIÓN .....  | 26        |
| 3.2 PROCEDIMIENTO DE BÚSQUEDA .....   | 28        |
| 3.2.1 Algoritmo: Obtener todas las ocurrencias que existan según el criterio de búsqueda en una base de datos.....  | 28        |
| 3.2.2 Algoritmo: Obtener todos los resultados de la base de datos una vez encontrado el criterio de búsqueda en el fichero almacenado.....                                | 29        |
| 3.2.3 Algoritmo: Obtener todas las ocurrencias que existan según el criterio de búsqueda en una base de datos creando y guardando los resultados en un fichero nuevo..... | 30        |
| 3.3 PERSONAL RELACIONADO CON EL SISTEMA. ....   | 31        |
| 3.4 FASE I: EXPLORACIÓN .....   | 32        |
| Historias de Usuario .....  | 32        |
| 3.5 FASE II: PLANIFICACIÓN.....   | 33        |
| Estimación de esfuerzos por HU .....  | 33        |
| Plan de Iteraciones .....   | 34        |
| Plan de duración de iteraciones .....   | 34        |
| 3.6 CONCLUSIONES PARCIALES .....  | 35        |

|  |           |
|--|-----------|
| <b>CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b> | <b>36</b> |
| 4.1 DISEÑO DE LA SOLUCIÓN PROPUESTA.....                       | 36        |
| 4.1.1 Tarjetas CRC.....  | 37        |
| REALIZARBUSQUEDA.....  | 37        |
| 4.2 TAREAS DE LA INGENIERÍA .....                              | 39        |
| 4.2.1 Iteración uno.....                                       | 39        |
| 4.3 PRUEBAS.....   | 41        |
| 4.3.1 Pruebas Unitarias .....                                  | 41        |
| 4.3.2 Pruebas de Aceptación.....                               | 47        |
| 4.4 CONCLUSIONES PARCIALES .....                               | 56        |
| <b>CONCLUSIONES.....</b>                                       | <b>57</b> |
| <b>RECOMENDACIONES.....</b>                                    | <b>58</b> |
| <b>BIBLIOGRAFÍA.....</b>                                       | <b>59</b> |
| <b>REFERENCIAS BIBLIOGRÁFICAS.....</b>                         | <b>60</b> |
| <b>ANEXOS .....</b>  | <b>62</b> |

## Figuras y Tablas

### ÍNDICE DE FIGURAS

|  |    |
|--|----|
| FIGURA 1: REPRESENTACIÓN DEL PROCESO DE BÚSQUEDA DE LA INFORMACIÓN (OBTENIDO DE:<br>HTTP://WWW.IDEANDALUCIA.ES)..... | 10 |
| FIGURA 2 FUNCIONAMIENTO SERVICIO WEB. (OBTENIDO DE: HTTP://WWW.W3C.ES) .....   | 11 |
| FIGURA 3 ESTRUCTURA DE UN MENSAJE SOAP. (OBTENIDO DE: HTTP://WWW.W3C.ES) .....                                       | 12 |
| ILUSTRACIÓN 4 ARQUITECTURA MODELO VISTA CONTROLADOR .....  | 27 |
| ILUSTRACIÓN 5 ESQUEMA FUNCIONAL DEL SERVICIO WEB.....  | 27 |
| ILUSTRACIÓN 6 ANEXO 1_ ESTRUCTURA BUNDLEBUSQUEDA .....   | 62 |
| ILUSTRACIÓN 7 ANEXO 2_ CASOS DE PRUEBA .....   | 63 |
| ILUSTRACIÓN 8 ANEXO 3_ RESULTADO DE LOS CASOS DE PRUEBA .....  | 63 |

### ÍNDICE DE TABLAS

|   |    |
|---|----|
| TABLA 1: RELACIÓN ENTRE LOS MÉTODOS DE BÚSQUEDA.....            | 16 |
| TABLA 2 DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y NO ÁGILES. .... | 19 |
| TABLA 3 PERSONAS RELACIONADAS CON EL SISTEMA. ....              | 31 |
| TABLA 4 HU BÚSQUEDA EN LA BASE DE DATOS.....                    | 33 |
| TABLA 5 ESFUERZO POR HISTORIAS DE USUARIO .....                 | 34 |
| TABLA 6 PLAN DE DURACIÓN DE LAS ITERACIONES .....               | 35 |
| TABLA 7 TARJETA CRC MOTORBUSQUEDACONTROLLER.....                | 37 |
| TABLA 8 TARJETA CRC CCONEXIONCONTROLLER .....                   | 38 |
| TABLA 9 TARJETA CRC CLASSRESULTADOCONTROLLER.....               | 38 |
| TABLA 10 HU A REALIZAR Y TIEMPO DE DURACIÓN. ....               | 39 |
| TABLA 11 TAREA DE INGENIERÍA #1, ITERACIÓN 1.....               | 39 |
| TABLA 12 TAREA DE INGENIERÍA #2, ITERACIÓN 1.....               | 40 |
| TABLA 13 TAREA DE INGENIERÍA #3, ITERACIÓN 1.....               | 40 |
| TABLA 14 TAREA DE INGENIERÍA #4, ITERACIÓN 1.....               | 40 |
| TABLA 15 TAREA DE INGENIERÍA #5, ITERACIÓN 1.....               | 41 |
| TABLA 16 CASO DE PRUEBA DE ACEPTACIÓN HU1_P1 .....              | 47 |
| TABLA 17 CASO DE PRUEBA DE ACEPTACIÓN HU1_P2.....               | 49 |
| TABLA 18 CASO DE PRUEBA DE ACEPTACIÓN HU1_P3.....               | 50 |

|   |    |
|---|----|
| TABLA 19 CASO DE PRUEBA DE ACEPTACIÓN HU1_P4..... | 51 |
| TABLA 20 CASO DE PRUEBA DE ACEPTACIÓN HU1_P5..... | 53 |
| TABLA 21 CASO DE PRUEBA DE ACEPTACIÓN HU1_P5..... | 54 |

## Introducción

La informatización es un proceso que se ha llevado a cabo a lo largo de los años en gran parte de las sociedades alrededor del mundo. Este proceso ha logrado en gran medida automatizar áreas importantes de los diferentes sectores ayudando a una mejor organización y disponibilidad de los datos. Una de las áreas a la que han llegado estos avances ha sido la geografía, con la implantación de nuevas tecnologías se ha ido dejando atrás poco a poco el uso de los mapas para darle paso a sistemas informáticos capaces de procesarlos, manipularlos y analizarlos.

Según **(Buzai, 2006)** “la geografía automatizada corresponde al proceso por el cual, a partir de iniciada la década de 1980, se incorporan progresivamente nuevos desarrollos en tecnología digital destinados al Análisis Espacial”. Este conjunto de tecnologías digitales entre las cuales se encuentran la Cartografía Computacional, el Procesamiento Digital y los SIG entre otros, presentan una gran variedad de aplicación, dándole paso a los SIG como tecnología de integración que se convierten estos en el principal medio para realizar un análisis socioespacial, transformándose así en una herramienta de múltiples posibilidades para el estudio, gestión y planificación territorial.

Un SIG se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar en bases de datos, analizar, transformar y presentar toda la información geográfica **(Brenes, 2005)**. Existe un punto intermedio entre la presentación y el almacenamiento de la información que no es más que la búsqueda de la misma, dicha búsqueda se realiza accediendo a una base de datos espacial. Existen variedades de métodos o técnicas para realizar la búsqueda, dichos métodos y técnicas varían de un SIG a otro y en algunos se pueden hallar más de una forma de búsqueda, por ejemplo: búsquedas por criterios: en donde hay parámetros anteriormente definidos dígame fechas, zonas geográficas, criterios temáticos; se encuentran también las búsquedas por catálogos: en donde se navega por una serie de categorías conformando estructuras jerárquicas de objetivos geográficos; por otra parte existen los buscadores de datos: permiten la exploración de la información residente en la base de datos, mediante búsquedas directas (por cadenas de caracteres). La búsqueda directa proporciona al usuario la opción de realizar búsquedas empleando filtros alfanuméricos (palabras clave, categorías y escalas) y/o temporales, en función de la fecha de creación de los datos. La búsqueda directa también puede estar dada solamente por cadenas de caracteres determinadas por el usuario y según su propio criterio.

Una de las funcionalidades más provechosas que proveen los SIG son los sistemas de búsqueda que incorporan. Las formas que implementan sus resultados en dependencia de la cantidad de información que almacenan, dígame cartografías con un nivel de complejidad alto en cuanto a datos, exigen a los desarrolladores crear soluciones que permitan un mejor y más rápido acceso a la información con la que requieran trabajar.

En Cuba, la Universidad de las Ciencias Informáticas (UCI) ha colaborado al proceso de informatización de la sociedad con el objetivo principal de disminuir la brecha digital existente en el país con respecto al mundo. Una de las áreas en la que se ha hecho un gran énfasis es en la Geografía aportando con el diseño e implementación de SIG. El departamento Geoinformática del Centro GEYSED (Geoinformática y Señales Digitales) es uno de los precursores en el desarrollo de estos sistemas los cuales son de diferentes temáticas, todos poseen un mismo propósito de contribuir en la toma de decisiones en el trabajo organizacional de cualquier entidad de negocio. Esto permite mediante sus funcionalidades brindar servicios de georeferenciación y localización de los recursos.

Los productos SIG que se han desarrollado en el Centro GEYSED comparten características similares en aspectos como la localización en los mapas de los objetivos geográficos partiendo de una búsqueda que se le realiza a las bases de datos espaciales. Dichas similitudes en la localización y búsqueda provocan las limitaciones sistémicas, anteriormente explicadas, que impiden realizar búsquedas por criterios definidos por quien vaya a utilizar el sistema. Esto viene dado porque los métodos de localización y búsqueda están generalmente definidos por parámetros ya establecidos por los desarrolladores y se utilizan técnicas como la búsqueda por criterios predeterminados, búsquedas por categorías. Debido a la generalización de esta situación es necesario encontrar una vía que posibilite la integración de algún tipo de herramienta que permita una búsqueda más flexible en cada uno de los Sistemas de Información Geográfica que se desarrollan en el Centro GEYSED.

Teniendo en cuenta las limitantes antes expuestas, se puede identificar como **problema a resolver** que *los procedimientos de búsqueda de objetivos geográficos en aplicaciones SIG del departamento Geoinformática presentan limitaciones sistémicas en la obtención de los resultados*, por lo que se define como **objetivo general** *desarrollar un Servicio Web que posibilite la búsqueda de objetivos geográficos en bases de datos espaciales de manera rápida y sencilla para los usuarios* delimitándose como **objeto de estudio** *los procesos asociados a las funcionalidades de búsqueda sobre bases de datos espaciales* y como **campo de acción** *la automatización de los procesos de búsqueda para base de datos espaciales*.

Por tanto se propone como **idea a defender** que *con la implementación de un Servicio Web que incluya algoritmos de búsqueda avanzados, se obtendrán resultados de manera más precisa, rápida y eficiente en las funcionalidades de localización en los Sistemas de Información Geográfica del Centro GEYSED.*

Para el cumplimiento del objetivo general se trazaron las siguientes tareas de investigación:

1. Caracterizar el estado del arte de los servicios Web, algoritmos de búsqueda y subsistemas de información geográfica para la consulta de datos espaciales.
2. Seleccionar y argumentar la Metodología de Desarrollo de Software a usar en el proceso.
3. Identificar y fundamentar las tecnologías idóneas para el desarrollo del sistema.
4. Identificar las principales funcionalidades de la herramienta.
5. Analizar y diseñar el servicio de búsqueda.
6. Implementar las funcionalidades del sistema.
7. Realizar pruebas a las funcionalidades de criticidad mayor del sistema.
8. Validar el resultado obtenido con la personalización de la herramienta para un negocio específico y sobre la base de una arquitectura orientada a servicios.

Con el correcto cumplimiento de las tareas se espera obtener los siguientes resultados o aportes:

1. Un servicio Web para búsqueda de información geográfica sobre bases de datos espaciales que integre sus resultados a un visor SIG.
2. Documentación teórica e ingenieril de la herramienta del buscador de objetos geográficos.

### **Métodos Científicos**

Durante todo el proceso investigativo se utilizarán un conjunto de métodos científicos. Estos métodos se clasifican en:

**Teóricos:** Posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado.



**Empíricos:** Estos métodos permiten extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de observaciones.

**Dentro de los teóricos se emplearán los siguientes:**

**Histórico-Lógico:** Se utilizará para el estudio de los procesos de gestión de información, trabajos e investigaciones anteriores, es decir para un mejor entendimiento y análisis del estado del arte lo que permitirá un eficaz desarrollo de la presente investigación.

**Analítico-Sintético:** Se define con el objetivo de analizar los diversos documentos relacionados con el proceso de desarrollo de los Sistemas de Información Geográfica y las metodologías existentes para la construcción de los artefactos necesarios así como para realizar una síntesis de los mismos.

**Dentro de los empíricos se empleará:**

**Observación:** Para realizar un registro visual de lo que ocurre en el entorno del problema para aportar nuevos elementos que puedan ser de interés científico y para observar cómo se realiza el proceso de desarrollo de herramientas de búsqueda en los Sistemas de Información Geográfica con el objetivo de lograr un mejor entendimiento de esta aplicación.

# Capítulo 1: Fundamentos Teóricos de la Investigación

## 1.1 Introducción

El este capítulo se plantean los principales elementos teóricos que serán de utilidad para lograr un mejor entendimiento de la investigación dígase definiciones y conceptos necesarios que fundamentarán todo lo que se plantee en este documento. Se analizan las funcionalidades de diferentes soluciones, principales tecnologías y herramientas a utilizar durante el desarrollo del sistema.

## 1.2 Conceptos asociados al dominio del problema

**Sistema de Información Geográfica (SIG):** Es una herramienta que brinda a las labores de uso y manejo de información geográfica toda la potencia de un ordenador, pues ha sido diseñada específicamente para trabajar con este tipo particular de información. Es decir un SIG debe permitir la lectura, edición, almacenamiento y, en términos generales, gestión de datos espaciales, también los análisis de dichos datos así como la generación de resultados tales como mapas, informes, gráficos **(Olaya, 2010)**.

Relacionado con los SIG existen diferentes elementos que dan vida al sistema entre ellos se encuentra el **Dato Espacial** el cual es una variable asociada a una localización del espacio. En diversos campos hay una necesidad por manejar información geométrica, geográfica o espacial, lo cual quiere decir que los datos guardan cierta relación con el espacio **(Guting, 2004)**.

Dato espacial es una variable asociada a una localización del espacio. Generalmente se hace uso de datos vectoriales expresados mediante tres tipos de objetos espaciales:

- Puntos: Se encuentran determinados por las coordenadas terrestres medidas por latitud y longitud.
- Líneas: Objetos abiertos que cubren una distancia dada y comunican varios puntos o nodos, aunque debido a la forma esférica de la tierra también se le consideran como arcos (carreteras, líneas telefónicas, etc.).
- Polígonos: Figuras planas conectadas por distintas líneas u objetos cerrados que cubren un área determinada (países, regiones, lagos, etc.).

Los datos espaciales también son caracterizados por su naturaleza georeferenciada y multidireccional. La primera indica que los datos están referenciados con respecto a la tierra. Por multidireccional se dice que existen diversas relaciones, es decir, todos se relacionan entre sí pero las relaciones no tienen el mismo peso. Para la utilización de estos datos los mismo son almacenados en **Base de datos Espaciales** las cuales son según **(Haithcoat, 1999)** "...una base de datos espacial es una colección de datos referenciados espacialmente, que actúan como un modelo de la realidad; en el sentido de que ella representa una serie o aproximación de fenómenos...". La representación digital puede ser de fenómenos del pasado, presente o futuro. Otros elementos significativos son los **Objetos Geográficos** los cuales no son más que abstracciones de elementos del mundo real que están asociadas a una posición geográfica y temporal definida, respectivamente, en un sistema de referencia espacial y temporal. Estos objetos presentes en el mundo real tienen dos tipos básicos de abstracción: ocurrencias y tipos **(ICDE, 2011)**.

**Ocurrencia o instancia:** Es cualquier objeto que es diferenciado de los demás teniendo en cuenta sus características propias, es decir, se particulariza o se vuelve único. Por ejemplo: Río Magdalena, Río Amazonas, Río Orinoco, etc. Se puede identificar cada objeto de manera inequívoca y con una ubicación espacial específica.

**Tipo:** Cuando un conjunto de elementos posee características similares y éstos son abstraídos como una clase de elemento, se les llama tipos. Esta forma de abstracción es comúnmente utilizada para la definición de catálogos. Deben ser documentados de tal manera que no dependan del producto o escala.

### 1.3 Conceptos asociados a la solución

**Servicio Web:** "Los servicios Web son sistemas de software que permiten el intercambio de datos y funcionalidad entre aplicaciones sobre una red. Esta soportado en diferentes estándares que garantizan la interoperabilidad de los servicios." **(Morales Machuca, 2008)**, según la World Wide Web Consortium lo define como "...un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene una interfaz descrita en un formato procesable por una máquina específicamente **WSDL** el cual es un tipo de documento XML que describe lo que hace un servicio Web, donde se encuentra y la forma de ser invocado. Este provee información muy importante para los desarrolladores, este lenguaje describe el formato de los mensajes que utiliza y a cuales puede responder **(Tidwell, 2002)**."

**XML** es uno de los lenguajes más utilizados para el intercambio de datos sobre la Web. Su desarrollo se remonta en el año 1996 por el grupo de trabajo de la World Wide Web Consortium lanzando su primera versión el 10 de Febrero de 1998 (**Van de Putte, 2004**). El lenguaje XML está concebido para describir objetos de datos llamados Documentos XML y describir de cierta forma los programas que los procesan. Está restringido bajo la norma ISO 8879 el Standar Generalized Markup Language.

Otros sistemas interactúan con el servicio Web en una manera prescrita por su descripción usando mensajes **SOAP** los cuales son un protocolo de la capa de aplicación para el intercambio de mensajes basados en XML sobre redes de computadores. Básicamente es una vía de transmisión entre un SOAP Sender y un SOAP Receiver, pero los mensajes SOAP deben interactuar con un conjunto de aplicaciones para que se pueda generar un “diálogo” a través de mensajes SOAP. Un mensaje SOAP es la unidad fundamental de una comunicación entre nodos SOAP (**Consortium, 2004**), típicamente enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la Web” (**W3C Consortium, 2004**).

## **1.4 Objeto de Estudio**

### **1.4.1 Procesos asociados a las funcionalidades de búsqueda sobre bases de datos espaciales**

Una de las funcionalidades básicas de los SIG es mostrar la información geográfica alojada en base de datos espaciales, este es un proceso en el cual intervienen una serie de entes que en su conjunto conforman un número de tecnologías que se han desarrollado con el tiempo para que la muestra de datos sea más eficiente y eficaz. Dichas bases de datos espaciales se almacenan en un Sistema Gestor de Base de Datos (SGBD) los cuales comprenden una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos.

Hoy en día existen varios modelos de bases de datos, las bases de datos relacionales, el modelo orientado a objetos, uno híbrido denominado modelo Objeto-Relacional y un concepto que ha tomado auge en los últimos años en el mundo de los SIG que es el de Geodatabase. Geodatabase se trata simplemente de una base de datos que almacena toda la información relativa a un conjunto de entidades espaciales (geometría, topología, identificadores, datos temáticos) (**Rigaux, et al., 2001**). Las ventajas de este modelo de trabajo son varias:

- Posibilidad de usar SQL, una versión ampliada de SQL en realidad para hacer consultas y análisis sobre mapas vectoriales.

- Mayor integración, en una sola herramienta, de todas las funciones para trabajar con información vectorial.
- El inconveniente es que se necesita un programa externo, o sea un SIG, para acceder a los datos y visualizarlos.

Entre los programas que permiten trabajar con Geodatabases caben destacar dos, en primer lugar Oracle Spatial y en segundo lugar PostgreSQL + PostGIS. Oracle posee como inconvenientes un elevado precio y PostgreSQL es una alternativa libre (y gratuita). PostGIS es una extensión, también libre, de PostgreSQL que le permite trabajar con geodatabases. PostGIS es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en el SIG. Un aspecto importante a tener en cuenta es que PostGIS ha sido certificado en 2006 por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas.

Partiendo de que se tiene la información en la base de datos se pueden realizar consultas que devuelva, utilizando las facilidades que otorga PostgreSQL + PostGIS, objetos espaciales (polígonos, líneas o puntos) para ser mostrados en el SIG. Pero en este proceso de visualización existe un módulo en específico que facilita la traducción por así decirlo entre el SIG y la base de datos así como de la base de datos para el SIG, que no es más que los servidores de mapas los cuales permiten transformar la información enviada a él para que sea entendible tanto por el SIG como por el SGBD.

En el entorno de los SIG los servidores de mapas deben ser interoperables; es decir, deben ser consultables por medio de unas especificaciones estandarizadas independientemente del servidor o cliente concreto que se use. Dichos estándares o especificaciones son desarrollados por organizaciones internacionales cuyo fin es la estandarización. Dichas organizaciones son la International Organization for Standardization (ISO) y Open Geospatial Consortium (OGC) las cuales son las encargadas del desarrollo de dichas estándares.

Existen una serie de ejemplos de servidores de mapas de código abierto los cuales son los más utilizados hoy en día, entre ellos se encuentra MapServer que entre sus características más destacadas está la utilización de varios lenguajes (PHP, Python, Perl) y la ejecución multiplataforma (Linux, Windows). Admite múltiples formatos de datos vectoriales tales como ESRI shapfiles, PostGIS. Además Mapserver cumple con una serie de especificaciones estipuladas por el OGC, dichas

estipulaciones son las siguientes: WMS (cliente/servidor) para producir mapas de datos referenciados espacialmente, WFS no transaccional (cliente/servidor), WMC, entre otros **(Andalucía, 2005)**.

Por otro lado se encuentra Geoserver que es un servidor de código abierto desarrollado en Java que permite a los usuarios compartir y editar datos geoespaciales. Geoserver soporta varios formatos de datos, incluyendo los provistos por PostGIS y Oracle Spatial. A través de protocolos estándares es capaz de generar archivos con formatos KML (del acrónimo en inglés Keyhole Markup Language), GML, Shapefile, y otros. Además es la implementación de referencia del Open Geospatial Consortium (OGC) para las normas Web Feature Service (WFS) y Web Coverage Service (WCS). Además Geoserver está certificado como servidor de alto rendimiento para Web Map Service (WMS) **(Andalucía, 2005)**.

Con la integración de todas estas tecnologías se obtiene como resultado final la visualización de los objetos espaciales en el SIG y de esta manera se completaría el proceso de búsqueda entre el SIG y la base de datos. El método para realizar la búsqueda de información espacial varía según las especificaciones por las que se desarrolle el SIG, entre ellos se pueden encontrar búsquedas por criterios, por catálogos, directas, todas basadas en el mismo funcionamiento explicado anteriormente. Dichos métodos se diferencian unos de otros en la facilidad de búsqueda que permiten y la libertad que brindan de especificar la búsqueda. Son algunos de ellos como la búsqueda directa que permite al usuario ser el que determine qué buscar libremente permitiéndole definir parámetros de búsqueda. **(Figura 1)**

En la actualidad en los SIG que se desarrollan en la UCI, específicamente en el Departamento Geoinformática del Centro GEYSED, las soluciones que se realizan están caracterizadas por el uso de métodos estáticos y poco manipulables desde el punto de vista de la facilidad de darle al usuario la libertad de que sea él quien decida qué buscar y cómo hacerlo. Estos métodos están dirigidos principalmente a búsquedas por parámetros y por catálogos, teniendo en cuenta que existe dicha limitante sistémica en los SIG desarrollados en el centro GEYSED es necesario plantearse una solución que permita la integración de un sistema de búsqueda directa a los diferentes SIG realizados en el centro. Una de las vías para realizar esta integración son los servicios Web los cuales se fundamentarán en el siguiente epígrafe.

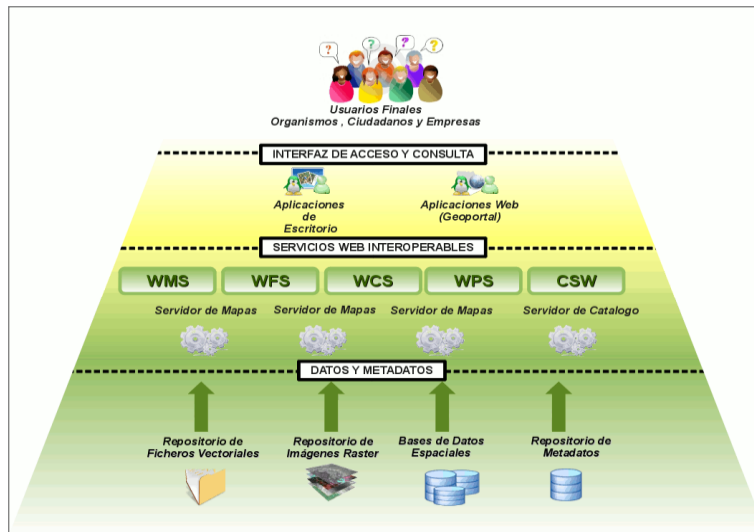
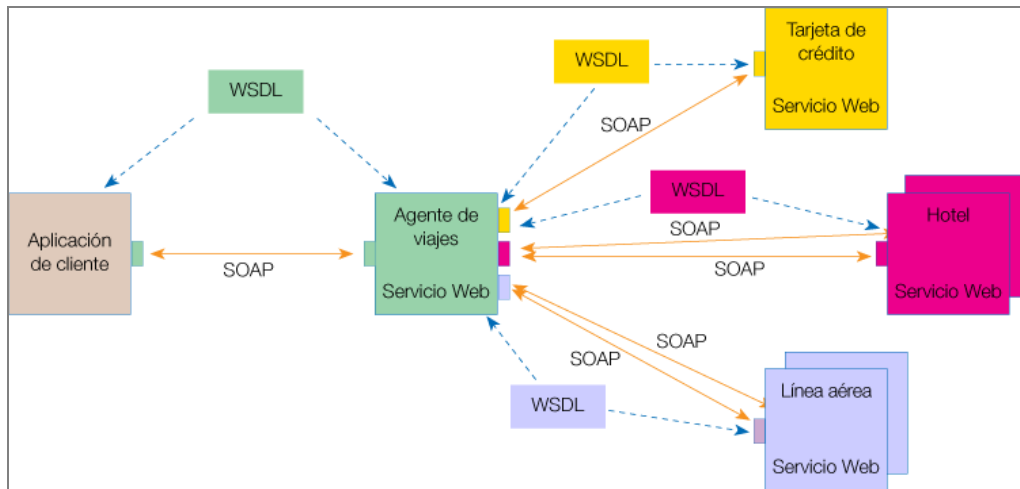


Figura 1: Representación del proceso de búsqueda de la información (Obtenido de: <http://www.ideandalucia.es>)

### 1.4.2 Servicios Web

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una definición adecuada que abarque todo lo que son e implican. Una posible será hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos o métodos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web (**Oficina Española W3C, 2010**). Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar. El funcionamiento de los mismos es en sí sencillo, según el ejemplo de la Figura 2 un usuario (que juega el papel de cliente dentro de los Servicios Web), a través de una aplicación, solicita información sobre un viaje que desea realizar haciendo una petición a una agencia de viajes que ofrece sus servicios a través de Internet. La agencia de viajes ofrecerá a su cliente (usuario) la información requerida.

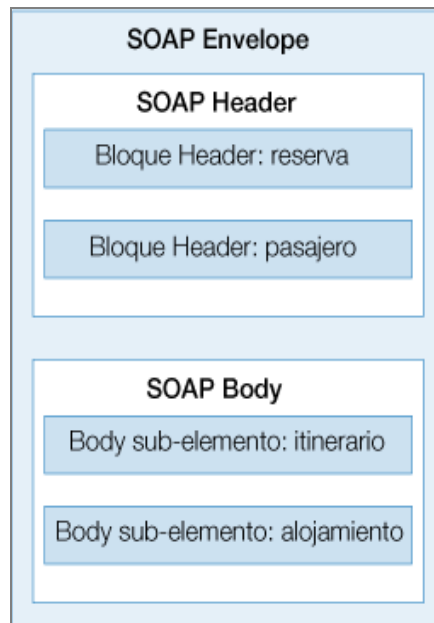


**Figura 2 Funcionamiento Servicio Web. (Obtenido de: <http://www.w3c.es>)**

Para proporcionar al cliente la información que necesita, esta agencia de viajes solicita a su vez información a otros recursos (otros Servicios Web) en relación con el hotel y la compañía aérea. La agencia de viajes obtendrá información de estos recursos, lo que la convierte a su vez en cliente de esos otros Servicios Web que le van a proporcionar la información solicitada sobre el hotel y la línea aérea. Por último, el usuario realizará el pago del viaje a través de la agencia de viajes que servirá de intermediario entre el usuario y el servicio Web que gestionará el pago.

En todo este proceso intervienen una serie de tecnologías que hacen posible esta circulación de información. Por un lado, estaría SOAP (Protocolo Simple de Acceso a Objetos). Se trata de un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP, SMTP. SOAP especifica el formato de los mensajes. El mensaje SOAP está compuesto por un *envelope* (sobre), cuya estructura está formada por los siguientes elementos: *header* (cabecera) y *body* (cuerpo).





**Figura 3 Estructura de un mensaje SOAP. (Obtenido de: <http://www.w3c.es>)**

Para optimizar el rendimiento de las aplicaciones basadas en Servicios Web, se han desarrollado tecnologías complementarias a SOAP, que agilizan el envío de los mensajes (MTOM) y los recursos que se transmiten en esos mensajes (SOAP-RRSHB).

Por otro lado, WSDL (Lenguaje de Descripción de Servicios Web), permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento procesable por dispositivos. WSDL representa una especie de contrato entre el proveedor y el que solicita. WSDL especifica la sintaxis y los mecanismos de intercambio de mensajes. Por último está UDDI que es un estándar abierto, que consiste en un catálogo donde se almacena información y descripción (documentos WSDL) de los servicios web que proveen las empresas, y cuyo sistema de comunicación está basado en mensajes SOAP. Es una especie de lista donde se puede publicar el servicio web (o buscar cualquier otro), para que pueda ser utilizado por el resto del mundo.

## **1.5 Análisis de otras soluciones existentes**

### **Métodos de búsqueda de Google Earth**

Google Earth es un programa informático similar a un Sistema de Información Geográfica (SIG), creado por la empresa Keyhole Inc., que permite visualizar imágenes en 3D del planeta Tierra, combinando imágenes de satélite, mapas que permiten ver imágenes a escala de un lugar específico

del planeta (**Autores, 2007**). Utiliza diferentes métodos de búsqueda y localización, uno de ellos es el motor de búsqueda de Google que se sirve como herramienta para el acceso a las bases de datos. En estas bases de datos es donde se almacena la información geográfica, la utilización del buscador brinda una gran flexibilidad de resultados de acuerdo con lo que se le pase como parámetro de búsqueda. Google Earth aprovecha las potencialidades de su motor de búsqueda e implementa diferentes métodos de realizarla en la aplicación. Entre los métodos se encuentran:

**Nombres exactos:** Conociendo el nombre exacto del negocio que se quiere buscar se procura introducir el nombre entre comillas para reducir el número de resultados coincidentes con la cadena entera. Este tipo de entrada es el más específico. Si se quitan las comillas se obtienen resultados más amplios o usando alguno de los otros métodos descritos en esta sección.

**Nombre parcial:** Conociendo parte del nombre (una palabra o más) del negocio que se busca, se puedes usar. Este tipo de búsqueda suele dar como resultado un número mayor de coincidencias que las búsquedas de nombre exacto.

**Palabras clave:** Las búsquedas de palabras clave dan como resultado las direcciones de los negocios del tipo indicado, aun cuando la propia palabra clave no aparezca en el nombre del negocio. Por ejemplo, la palabra clave peluquería puede devolver resultados como Estética Mario o Salón de belleza Rulos y otros negocios que estén clasificados como peluquerías.

### **Métodos de búsqueda de la IDE-CHG**

La IDE-CHG (Infraestructura de Datos Espaciales de la Confederación Hidrográfica de Guadalquivir) el cual aporta una serie de servicios como son un visor cartográfico, servicios de descarga, teledetección. Entre los métodos de búsqueda que posee están:

**Buscador de Metadatos:** Se trata de un cliente de búsqueda que ofrece la posibilidad de localizar conjuntos de datos geográficos disponibles a una escala determinada, de una zona particular, sobre un tema específico, y en una fecha o rango de fechas.

El buscador de metadatos está dividido en dos secciones: **búsquedas directas** y **búsquedas por categorías**. La primera proporciona al usuario la opción de realizar búsquedas empleando filtros alfanuméricos (palabras clave, categorías y escala) y/o temporales, en función de la fecha de creación, edición, publicación, etc. de los datos. Por su parte, la búsqueda por categorías ordena los metadatos

de la información en grupos temáticos, permitiendo ver la información contenida en dichos grupos. En ambos casos el resultado de la búsqueda será un listado con los metadatos encontrados.

**Búsqueda por Categorías:** La pestaña 'Categorías' contiene un listado con las ocho categorías que almacenan los metadatos de la información geográfica disponible. Las categorías son:

- Cartografía base.
- Infraestructuras hidráulicas.
- Marco administrativo y de gestión.
- Masas de agua.
- Medio físico.
- Redes de control.
- Usos del suelo.
- Zonas protegidas.

Para desplegar cada una de las categorías y visualizar el contenido, es necesario pulsar sobre el nombre de la categoría que nos interese.

**Búsquedas Directas:** La pestaña 'Directa' permite realizar búsquedas empleando filtros alfanuméricos y/o temporales mediante una serie de opciones que el usuario puede configurar. Por un lado, los filtros alfanuméricos ofrecen la posibilidad de realizar búsquedas mediante palabras clave, categorías y/o escala; por otro, el filtro temporal permite buscar metadatos entre dos fechas seleccionadas por el usuario.

**Búsqueda por Topónimos:** El buscador de topónimos es una herramienta que permite al usuario localizar espacialmente elementos cartográficos a través de una consulta según distintos criterios de búsqueda. La ventana principal de la herramienta está dividida en dos zonas principales. La parte izquierda, donde se presentan los diferentes criterios de búsqueda y la derecha, reservada para mostrar el resultado de la búsqueda.

## **Métodos de búsqueda de QGIS**

QGIS es un Sistema de Información Geográfica de código abierto que fue concebido inicialmente con el objetivo de proporcionar a los usuarios un visor de datos GIS. QGIS ha alcanzado el punto en su evolución entre los años 2009 y la actualidad y está siendo utilizado por un gran número de usuarios con necesidades de visualización en sus datos diarios GIS. QGIS soporta un número de raster y formatos de datos vectoriales. Su sistema de búsqueda se basa principalmente en la búsqueda de módulos de datos específicos y en la búsqueda por categorías y atributos. (Sherman, 2004-2007)

## **Módulo de búsqueda de la Plataforma de Servicios de Mapas de la UCI**

Es una plataforma desarrollada en el Centro GEYSED para brindar servicios de mapas mediante el estándar WMS e incluye funcionalidades de localización todos basados en parámetros predeterminados entre ellos están:

1. **Localizar Edificio Residencial:** Devuelve un componente de mapa con la localización geográfica de un edificio dado su número.
2. **Localizar Persona Por CI:** Devuelve un componente de mapa con la localización geográfica de la ubicación residencial de una persona dado su número de carné de identidad.
3. **Localizar Persona Por Solapín:** Devuelve un componente de mapa con la localización geográfica de la ubicación residencial de una persona dado su número de solapín.
4. **Localizar Persona Por Usuario:** Devuelve un componente de mapa con la localización geográfica de la ubicación residencial de una persona dado su usuario.
5. **Localizar Computadora:** Devuelve un componente de mapa con la localización geográfica de una computadora dado su número de IP.
6. **Localizar Objetivo:** Devuelve un componente de mapa con la localización geográfica de un objetivo socioeconómico dado su nombre.

**(Corchete, 2011)**

## **Métodos de búsqueda de SIGUCI**

Aplicación desarrollada en la UCI por el proyecto Aplicativos SIG la cual es una solución creada para la georreferenciación de los objetivos socioeconómicos del centro y la toma de decisiones sobre el entorno geográfico del mismo. Dispone para los usuarios un conjunto de funcionalidades de representación, análisis temático y edición sobre el mapa digital de la universidad, que se sirve desde una biblioteca de datos espaciales. Sus funcionalidades de localización se basan en búsqueda por parámetros predefinidos. **(Aplicativos SIG, 2011)**

## **IDERC**

La Infraestructura de Datos Espaciales de la República de Cuba (IDERC) abarca las políticas, tecnologías, estándares y recursos humanos necesarios para la efectiva recolección, administración, acceso, entrega y utilización de los datos espaciales a nivel nacional en función de la toma de decisiones económicas, políticas y sociales, y del desarrollo sostenible.

La IDERC permitirá compartir la información geográfica en un ambiente cooperativo interinstitucional para soportar la toma de decisiones sociales, económicas y ambientales. El Portal Geoespacial Nacional o GeoPortal es un punto de acceso nacional a la información geográfica disponible en la Infraestructura de Datos Espaciales de la República de Cuba (IDERC, 2004).

Los datos fundamentales hasta el momento son: Distribución Política Administrativa, Hidrografía, Puntos Poblados y asentamientos urbanos, Elevaciones, Viales y Nombres Geográficos. El visor de mapas de la IDERC posee búsquedas por catálogos como método de búsqueda conformando así estructuras jerárquicas de objetivos geográficos.

Los primeros servicios del GeoPortal son un Diccionario Geográfico accesible mediante mapas interactivos y los Mapas de Población del Anuario Estadístico de Cuba editado en el 2003. El lanzamiento del GeoPortal tuvo lugar el 30 de Noviembre de 2004 con carácter experimental. Posee técnicas de búsqueda basada en parámetros, además de navegación por el mapa (IDERC, 2004).

## 1.6 Conclusión del estudio de las Soluciones

Luego de analizadas las soluciones expuestas anteriormente se observa la existencia de una similitud en la mayoría de los SIG estudiados en cuanto a los métodos de búsqueda que utilizan. En la mayoría de los casos se muestra una tendencia a utilizar métodos de búsqueda estática y poco flexibles, es decir, que están determinados por parámetros predefinidos como se muestra en la **Tabla 1**. Esto trae consigo la dependencia del usuario de realizar la búsqueda a partir de lo que se encuentra ya hecho. El caso de Google Earth utiliza como herramienta principal su motor de búsqueda que permite una amplia gama de resultados, brindándoles a los usuarios gran cantidad de información de cada resultado obtenido. Realizar una herramienta que permita una búsqueda determinada por el usuario y no por especificaciones dadas por los desarrolladores o el sistema permitirá mostrar información más detallada, además de una variedad más amplia de resultados dándole la oportunidad al usuario de decidir el mismo que localizar.

**Tabla 1: Relación entre los métodos de búsqueda.**

| Nombre       | Software Libre | Búsqueda Catálogos | Búsqueda Criterios | Búsqueda Datos |
|--------------|----------------|--------------------|--------------------|----------------|
| Google Earth | NO             | X                  | X                  | X              |
| PLASMA       | SI             | -                  | X                  | -              |

|               |    |   |   |   |
|---------------|----|---|---|---|
| <b>SIGUCI</b> | SI | X | X | - |
| <b>IDERC</b>  | SI | X | X | - |
| <b>IDECHG</b> | NO | X | X | - |

## 1.7 Conclusiones Parciales

En el desarrollo del capítulo se abordaron diferentes conceptos importantes, se estudiaron algunas de las principales soluciones existentes para lograr un mejor entendimiento y una visión de la solución del problema a resolver. Ello permite tener un mejor enfoque del objetivo que se desea lograr, así como centrarse en el estudio de temáticas de importancia que den al traste con la investigación a realizar. Teniendo en cuenta lo antes expuesto están sentadas las bases para determinar cuáles tecnologías utilizar para comenzar el sistema a desarrollar.

## Capítulo 2: Análisis de las tecnologías y herramientas a utilizar

### 2.1 Introducción

En este capítulo se explica mediante una descripción detallada las herramientas y tecnologías a utilizar para el desarrollo de la herramienta de búsqueda, así como las principales ventajas que estas ofrecen.

### 2.2 Metodología de desarrollo de software

El desarrollo de *software* no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte existen aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto.

Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto *software*. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas (**Newkirk, 2001**). Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir *software*, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

#### 2.2.1 Metodologías de desarrollo Ágiles

Las metodologías ágiles son sin duda uno de los temas recientes en la ingeniería de *software* que están acaparando gran interés. En la actualidad se está viviendo con intensidad un debate abierto

entre los partidarios de las metodologías tradicionales (referidas peyorativamente como "metodologías pesadas") y aquellos que apoyan las ideas emanadas del "Manifiesto Ágil"<sup>1</sup>.

Por un lado, para muchos equipos de desarrollo el uso de metodologías tradicionales les resulta muy lejano a su forma de trabajo actual considerando las dificultades de su introducción e inversión asociada en formación y herramientas. Por otro, las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas se ajustan a un amplio rango de proyectos industriales de desarrollo de *software*; aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías.

**Tabla 2 Diferencias entre metodologías ágiles y no ágiles.**

| <b>Metodologías Ágiles</b>   | <b>Metodologías Tradicionales</b>  |
|--|--|
| Basadas en heurísticas provenientes de prácticas de producción de código | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |
| Especialmente preparados para cambios durante el proyecto                | Cierta resistencia a los cambios   |
| Impuestas internamente (por el equipo de desarrollo)                     | Impuestas externamente   |
| No existe contrato tradicional o al menos es bastante flexible           | Existe un contrato prefijado   |
| El cliente es parte del equipo de desarrollo                             | El cliente interactúa con el equipo de desarrollo mediante reuniones               |
| Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio         | Grupos grandes y posiblemente distribuidos   |
| Pocos artefactos   | Más artefactos   |
| Pocos roles  | Más roles  |
| Menos énfasis en la arquitectura del software                            | La arquitectura del software es esencial y se expresa mediante modelos             |

<sup>1</sup> agilemanifesto.org



## 2.2.2 Programación Extrema

La metodología de desarrollo seleccionada fue XP. XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. La metodología XP se centra en el trabajo en equipo, contribuye a lograr una buena comunicación entre el cliente y el equipo de desarrollo, trayendo consigo un mejor clima en el área de trabajo; instruye a sus desarrolladores pues una de sus principales tareas es el aprendizaje de los mismos. XP se fundamenta de una retroalimentación continua entre el equipo de trabajo y el cliente. **(Beck, 2000)**

Las características esenciales de XP son las siguientes: historias de usuario (HU), roles, proceso y prácticas. **(Joskowicz, 2002)**

**HU:** es la técnica utilizada para especificar las necesidades del *software*. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean funcionales o no funcionales. El tratamiento de las HU es muy dinámico y flexible. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

### **Roles:**

- Programador: el programador escribe las pruebas unitarias y produce el código del sistema.
- Cliente: escribe las HU y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las HU y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- Encargado de pruebas: ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- Encargado de seguimiento: proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, con el objetivo de mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- Entrenador: es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas de XP y se siga el proceso correctamente.

- Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto.
- Gestor: es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es la coordinación.

**Proceso:** el ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al principio.
- El ciclo de vida ideal de XP consiste de seis fases:
  - Exploración.
  - Planificación de entrega (*Release*).
  - Iteraciones.
  - Producción.
  - Mantenimiento.
  - Muerte del Proyecto.

**Prácticas:** la principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione.

El objetivo de XP es muy simple: la satisfacción del cliente. Esta metodología trata de dar al cliente el *software* que él necesita y en el momento que lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. XP se encarga de potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y los desarrolladores, son parte del equipo y están involucrados en el desarrollo de *software*.

XP mejora un proyecto de *software* de cuatro formas esenciales: comunicación, simplicidad, retroalimentación y coraje. Una de las características de la programación extrema es que es imposible prever todo antes de comenzar a programar; es imposible o si lo fuera es demasiado costoso e innecesario, ya que muchas veces se gasta demasiado tiempo y recursos en cambiar la

documentación de la planificación para que se parezca al código. Para evitar esto, XP intenta implementar una forma de trabajo donde se adapte fácilmente a las circunstancias. **(Beck, 2000)**

XP se adapta al modelo de trabajo seleccionado, ya que en XP existe un conjunto de prácticas que si se llevan a cabo se lograrán los objetivos trazados. Por una parte el cliente se encuentra en todo momento en contacto con los desarrolladores del software, otra de las prácticas son las pequeñas y frecuentes entregas lo que permite un control constante de las tareas y en cortos plazos de tiempo se pueden tener resultados. En XP está orientado además a proyectos pequeños y es una metodología que siempre espera cambios a lo largo del desarrollo del proyecto.

### **2.3 Marco de trabajo**

Symfony 2 es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones Web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes como conexión a bases de datos, seguridad de aplicaciones, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web. Además Symfony2 posee una gran independencia en los módulos que se van creando a medida que se desarrolla un sistema. El propio núcleo del *framework* está dividido en módulos que tienen una alta cohesión, lo que permite la reutilización de los mismos fuera de un proyecto basado en Symfony2. Esto permite hacer cualquier desarrollo Web utilizando dichos módulos – o Bundles como se llaman en Symfony2 – sin tener por qué usar el *framework* por completo.

Además, el gran esfuerzo que han hecho en la seguridad del propio *framework* – utilizando técnicas para evitar inyección SQL, XSS e incluso CSRF – así como la integración de PHP-Unit para la realización de pruebas unitarias o el uso de las últimas ventajas del futuro PHP6 como pueden ser los *namespaces*, abren un gran abanico de posibilidades para el desarrollador Web.

Symfony2 está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. Symfony2 es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Además Symfony2 hace uso de una arquitectura MVC para sus aplicaciones la cual provee una mayor

seguridad y organización en los proyectos desarrollados. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux) como en plataformas Windows. **(Symfony, 2012)**

La utilización de su versión 2 en el desarrollo del servicio de búsqueda viene encaminada a que el mismo será integrado como una funcionalidad más a la plataforma servicio de mapas PLASMA, el cual en un futuro cercano se migrará completamente a esta versión de Symfony. También se tiene en cuenta las futuras proyecciones del Proyecto Aplicativos SIG <sup>2</sup> del Centro GEYSED de la UCI en tomar esta versión de Symfony como su herramienta principal de trabajo y otra razón importante es la escalabilidad que proporciona, lo que permite extender el margen de operaciones sin perder calidad además de manejar el crecimiento continuo de trabajo de manera fluida.

## 2.4 Marco de trabajo para Servicios Web

El desarrollo de Servicios Web usando la biblioteca nativa de PHP o NuSOAP no cumple con los estándares que promueven interoperabilidad entre plataformas debido a que el consumo de servicios Web que brindan estas librerías trae consigo limitaciones desde el punto de vista de la integración en aplicaciones que no se hayan desarrollado completamente en PHP. Esto viene dado a que PHP no define tipos de datos como otros lenguajes y el estilo utilizado, rpc, para el desarrollo de servicios Web usando la biblioteca de PHP o NuSOAP no identifica los tipos de datos.

Para resolver este problema la empresa WSO2<sup>3</sup> ha creado un *framework* (WSF/PHP) que se encuentra en su versión 2.1.0 que permite implementar de forma muy similar a los *frameworks* anteriores servicios Web en PHP permitiendo la identificación de los tipos de datos en cualquiera que sea el lenguaje a través del cual se consuma el servicio Web.

Algunas de sus funcionalidades más elementales:

- Generación del WSDL a partir del código en PHP y viceversa.
- Exponer funciones o clases como servicios Web.
- Generación a partir del WSDL del código tanto del cliente como del servidor.
- Implementación de seguridad usando ws-security.
- Envío de adjuntos.
- Servicios de acceso a datos.

---

<sup>2</sup> Proyecto con la misión del desarrollo de aplicaciones SIG personalizadas sobre la base la Plataforma GeneSIG en un esquema de trabajo basado en líneas de productos de software.

<sup>3</sup> <http://wso2.com/>

- Interoperabilidad con plataformas como JAVA y .NET.

## **(WSO2)**

Por otra parte algunas de las ventajas de su uso están en que la licencia que posee es la de Apache v2.0 permitiendo al usuario del software la libertad de usarlo para cualquier propósito. Esto permite distribuirlo, modificarlo, y distribuir versiones modificadas de ese software, además cumple con todos los estándares en cuanto al formato (XML) en que los datos se vayan a intercambiar, así como protocolos (UDDI) para la publicación de los servicio Web, implementa muy bien los aspectos de seguridad a nivel de servicios Web, es muy fácil de usar y configurar ya que no hay que no hay que programar nada solo configurar el XML.

El uso del marco de trabajo en el desarrollo del servicio de búsqueda viene orientado a la futura integración que tendrá el mismo con la plataforma de servicios de mapas PLASMA. Esta integración condiciona al servicio de búsqueda al uso de tecnologías, como Symfony2 explicado anteriormente, el lenguaje de programación PHP y WSF/PHP, para así lograr la compatibilidad necesaria.

## **2.5 Lenguaje de Programación**

El lenguaje de programación escogido es PHP en su versión 5.3, pues por sus características se considera que posee la mejor mezcla entre flexibilidad y rendimiento para la realización de páginas web dinámicas. Además cuenta con abundante documentación lo que permite un mayor entendimiento del mismo. Este lenguaje está asociado con una extensa biblioteca que va creciendo conforme se realizan nuevas versiones, la cual permite la realización de disímiles tareas como el acceso a base de datos y el tratamiento de ficheros. Es un lenguaje abierto, multiplataforma, presenta un amplio soporte para la Programación Orientado a Objetos. Además soporta varios SGBD, como MySQL, Oracle y PostgreSQL y es atendido por una amplia comunidad de desarrolladores que lo actualizan frecuentemente con extensiones y mejoras y además encuentran y reparan errores de funcionamiento del lenguaje. Por último una característica importante es que PHP es el lenguaje nativo de los *frameworks* a utilizar para el desarrollo del servicio. **(PHP, 2012)**

## **2.6 Entorno de Desarrollo**

Para la realización del sistema se utiliza como entorno de desarrollo Netbeans en su versión 7.0.1 siendo consecuentes con el lenguaje y frameworks previamente propuestos, pues brinda la posibilidad de desarrollar aplicaciones web utilizando PHP5 y ofrece soporte para Symfony. Además es un proyecto de código abierto sin restricciones para su uso. Permite desarrollar las aplicaciones mediante

módulos, lo que facilita la posterior extensión de dichas aplicaciones. A pesar de estar escrito en Java puede utilizarse para escribir, compilar, depurar y ejecutar programas en otros lenguajes de programación como es el caso de PHP. Ofrece autocompletado de código, marcado de error para el lenguaje PHP y tiene un potente debugger integrado, con facilidad de utilización. El proyecto de NetBeans está apoyado por una vibrante comunidad de desarrolladores y ofrece una amplia documentación y formación de recursos, así como una variada selección de terceros plug-ins. **(Netbeans, 2012)**

## **2.7 Sistema Gestor de Bases de Datos (SGBD)**

Para almacenar la información que se manipula en un gran porcentaje de aplicaciones, se utilizan los SGBD, programas que garantizan la integridad y seguridad de la información y que sirven como intermediarios entre las aplicaciones y los datos. El gestor de base de datos elegido fue PostgreSQL V8.4, pues permite el soporte para datos espaciales en un SIG. Tiene soporte nativo para los lenguajes más populares del medio, como PHP, seleccionado anteriormente. Es una herramienta soberana, con una gran comunidad de usuarios que permite su continuo desarrollo. Cuba cuenta con una comunidad de desarrollo integrada por profesionales y estudiantes de la UCI, lo que constituye un factor importante en cuanto a fuente de información se refiere.

PostgreSQL funciona en la mayoría de los Sistemas Operativos más utilizados incluyendo, Linux, varias versiones de UNIX y Windows por lo que es un sistema multiplataforma, y utiliza principalmente como lenguaje de consulta a la base de datos SQL (Structured Query Language). **(Group, 1996)**

## **2.8 Conclusiones Parciales**

En el capítulo presentado se realiza un estudio de los principales conceptos y características de las herramientas y tecnologías a utilizar en la modelación e implementación del servicio Web de búsqueda de objetos geográficos. Se selecciona como metodología de desarrollo XP permitiendo un desarrollo centrado en la relación con el cliente ya que este se encuentra siempre en contacto con el desarrollador. De igual forma, la utilización de los *frameworks* seleccionados permite una mayor flexibilidad y usabilidad del sistema así como un desarrollo más dinámico junto con PHP como lenguaje nativo de las mismas y seleccionado para el desarrollo.

## Capítulo 3: Descripción de la solución propuesta

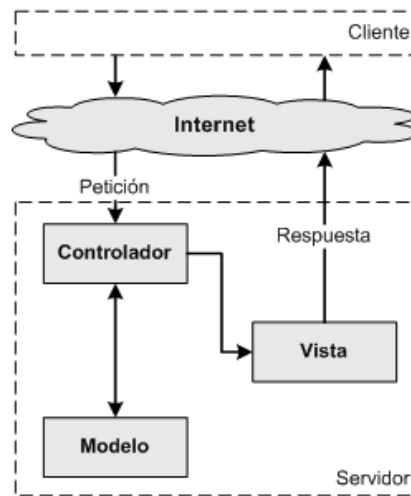
En este capítulo se describen las principales características del sistema, se hace mención a las fases de exploración y planificación, propias de la metodología ágil de desarrollo utilizada para la implementación del sistema. Además se muestran las historias de usuarios que fueron escritas por el cliente y se definen las cuatro iteraciones en la que se va a implementar el sistema.

### 3.1 Descripción de la solución

El departamento de Geoinformática del Centro GEYSED, tiene entre sus tareas la creación de SIG de diferentes temáticas. Cada uno de estos sistemas comparte características similares en cuanto a los métodos o herramientas de búsqueda que posee para la localización de información geográfica. Un SIG para realizar la búsqueda de datos en una base de datos espacial puede utilizar varios métodos desde una búsqueda por criterio, hasta una búsqueda por catálogos. Esto significa que los métodos de búsquedas son estáticos y poco flexibles debido a que están determinados por un parámetro predefinido por el sistema o por quien lo haya desarrollado.

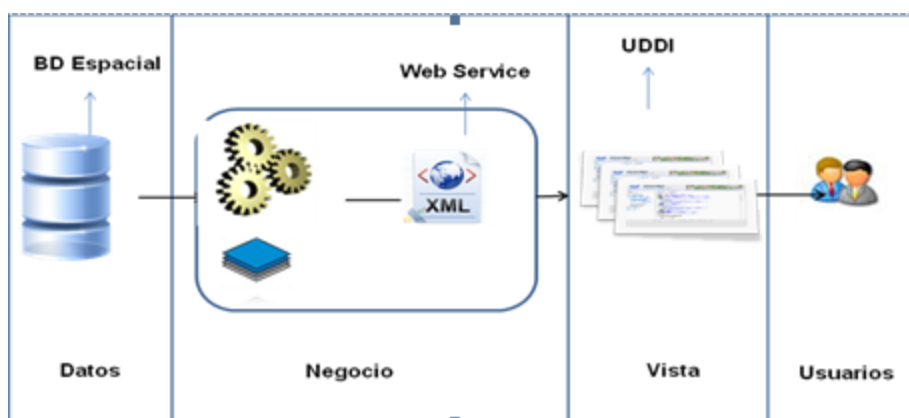
En este trabajo se propone la creación de un servicio Web de búsqueda que se integre a cada SIG para facilitar la obtención de resultados por parte del usuario del SIG y una mayor flexibilidad a la hora de realizar búsquedas sobre un objetivo en especial. Para el desarrollo del sistema se propone la creación de un módulo de búsqueda el cual será desarrollado utilizando las tecnologías seleccionadas previamente. Dicho módulo formará parte de la Plataforma de Servicios de Mapas PLASMA (Corchete, 2011) aportando así un servicio más a dicha solución.

El acceso a este módulo será a través de un servicio Web, cuya representación gráfica se observa en la **Ilustración 5**, al que los usuarios o sistemas que lo consuman podrán hacerlo a través de una vista desde donde llamarán a la función remota que se encargará de manipular los datos entrados así como los resultados obtenidos. Esta función remota estará vinculada a una serie de algoritmos, que se presentarán más adelante, permitiendo la interacción con la base de datos en donde esta almacenada la información. El módulo cuenta con una arquitectura Modelo Vista Controlador (**Ilustración 4**) definida por el *framework* Symfony2 en el cual se desarrolla.



**Ilustración 4 Arquitectura Modelo Vista Controlador**

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.



**Ilustración 5 Esquema funcional del servicio Web**



## 3.2 Procedimiento de búsqueda

El eje principal de este servicio Web es la búsqueda en la base de datos sin saber de cual se trate, además de que se integre a los SIG desarrollados en el Centro GEYSED. Para lograr esto se realiza una búsqueda en la base de datos que esté conectado el servicio, definiendo los campos en los que se va a buscar. Partiendo de allí se van realizando una serie de operaciones hasta obtener la consulta SQL que brindará la información buscada. Para este proceso se definen una serie de algoritmos que permiten realizar esta función además de optimizar la búsqueda para lograr hacerla más rápida.

### 3.2.1 Algoritmo: Obtener todas las ocurrencias que existan según el criterio de búsqueda en una base de datos.

---

**Entrada:** Criterio de búsqueda el cual puede ser una palabra o parte de ella o un número, además de tres valores que pueden ser o no nulos y serán los que se mostrarán como resultado.

**Salida:** Todas las ocurrencias encontradas en la base de datos con información adicional que será escogida por el desarrollador.

1. Abrir la conexión a la base de datos.
2. Crear variable columna.
3. Verificar SI existe el fichero en el cual se guardan las búsquedas realizadas.
  - 3.1. De existir el mismo se guarda su contenido en un arreglo.
  - 3.2. **Para toda** posición del arreglo obtener cada palabra que este guarda.
    - 3.2.1. Verificar que el criterio de búsqueda sea igual a algunos de los valores guardados en el arreglo.
      - 3.2.1.1. Si se encuentra alguna coincidencia acceder al método `devolverDatosEncontradosEnFichero`.
      - 3.2.2. De no ser así acceder al método `devolverDatosGuardandoEnFichero`.

#### 3.3. Fin Para

4. Si no existe el fichero donde se guardan las búsquedas realizadas se procede a crear y guardar en un nuevo fichero nuevo la búsqueda realizada.
5. Luego se crea otro fichero donde se guardarán las consultas que devuelvan los resultados válidos.
6. Almacenar en la variable `devolverSchemasTabla` la consulta que devolverá cada uno de los esquemas de la base de datos acompañado de las tablas propias de cada esquema.
7. Realizar consulta anterior a la base de datos.

8. **Para toda** posición de arreglo resultante de la consulta anterior obtener de cada posición schemaname y tablename.
  - 8.1. Almacenar en la variable nombreColumna la consulta que devolverá los nombres de las columnas de la tablename.
  - 8.2. Realizar consulta anterior a la base de datos.
  - 8.3. **Para toda** posición del arreglo resultante de la consulta anterior obtener de cada posición colum\_name.
    - 8.3.1. Verificar para cada colum\_name la existencia de la palabra nombre, name o una palabra similar al nombre de la tabla, ya que esto es lo que se compara con la variable búsqueda.
  - 8.4. **Fin Para.**
  - 8.5. Se van verificando el contenido de los valores de entrada al algoritmo para así saber cuáles de estos valores son nulos o no y así construir la consulta que devolverá los resultados.
  - 8.6. Almacenar en la variable devolverValoresPorTabla la consulta final que devolverá la información solicitada esta consulta debe contener los valores almacenados en las variables schemaname, tablename y colum\_name además del valor almacenado en la variable búsqueda y los valores que no son más que información adicional que se obtendrá de la base de datos.
  - 8.7. Realizar consulta anterior a la base de datos.
  - 8.8. **Para toda** posición del arreglo resultante de la consulta anterior obtener de cada posición los valores que se solicitados.
  - 8.9. Guardar en cada posición del arreglo resultado un objeto de la clase classResultado cada uno de estos valores.
  - 8.10. **Fin Para.**
9. **Fin Para.**
10. Retornar arreglo resultado.

---

### **3.2.2 Algoritmo: Obtener todos los resultados de la base de datos una vez encontrado el criterio de búsqueda en el fichero almacenado.**

---

**Entrada:** Criterio de búsqueda el cual puede ser una palabra o parte de ella o un número, además de 3 valores que pueden ser o no nulos y serán lo que se mostrará como resultado.

**Salida:** Todas las ocurrencias encontradas en la base de datos con información adicional que será escogida por el desarrollador.

1. Se obtienen todos los valores del fichero donde están guardadas las consultas que en su momento devolvieron los resultados deseados y se guardan en un arreglo.
2. **Para cada** posición del arreglo se obtienen las consultas.
  - 2.1. Se realiza la consulta a la BD obtenida y se guarda el resultado en la variable result.
  - 2.2. Se divide la consulta por los espacios que esta posea.
  - 2.3. Se obtiene el nombre de la columna name o nombre.
    - 2.3.1. **Para cada** posición del arreglo result se obtienen sus valores.
    - 2.3.2. Se verifica la cantidad de propiedades de cada valor de arreglo de objetos y dependiendo de la cantidad se crea un objeto resultado con los parámetros adecuados.
    - 2.3.3. **Fin Para**

### 3. Fin Para

---

**3.2.3 Algoritmo: Obtener todas las ocurrencias que existan según el criterio de búsqueda en una base de datos creando y guardando los resultados en un fichero nuevo.**

---

**Entrada:** Criterio de búsqueda el cual puede ser una palabra o parte de ella o un número, además de 3 valores que pueden ser o no nulos y serán lo que se mostrará como resultado.

**Salida:** Todas las ocurrencias encontradas en la base de datos con información adicional que será escogida por el desarrollador.

1. Se guarda la búsqueda realizada.
2. Luego se crea otro fichero donde se guardarán las consultas las cuales devuelva los resultados válidos.
3. Almacenar en la variable devolverSchemasTabla la consulta que devolverá cada uno de los esquemas de la base de datos acompañado de las tablas propias de cada esquema.
4. Realizar consulta anterior a la base de datos.
5. **Para toda** posición de arreglo resultante de la consulta anterior obtener de cada posición schemaname y tablename.
  - 5.1. Almacenar en la variable nombreColumna la consulta que devolverá los nombres de las columnas de la tablename.

5.2. Realizar consulta anterior a la base de datos.

5.3. **Para toda** posición del arreglo resultante de la consulta anterior obtener de cada posición `column_name`.

5.3.1. Verificar para cada `column_name` la existencia de la palabra nombre, `name` o una palabra similar al nombre de la tabla, ya que esto será lo que se compara con la variable `búsqueda`.

5.4. **Fin Para.**

6. Se van verificando el contenido de los valores de entrada al algoritmo para así saber cuáles de estos valores son nulos o no y así construir la consulta que devolverá los resultados.
7. Almacenar en la variable `devolverValoresPorTabla` la consulta final que devolverá la información solicitada esta consulta debe contener los valores almacenados en las variables `schemaname`, `tablename` y `column_name` además del valor almacenado en la variable `búsqueda` y los valores que no son más que información adicional que se obtendrá de la base de datos.
8. Realizar consulta anterior a la base de datos.
  - 8.1. **Para toda** posición del arreglo resultante de la consulta anterior obtener de cada posición los valores que se solicitados.
9. Guardar en cada posición del arreglo resultado un objeto de la clase `classResultado` cada uno de estos valores.
  - 9.1. **Fin Para.**
10. **Fin Para.**
11. Retornar arreglo resultado.

### 3.3 Personal relacionado con el sistema.

Para determinar al personal relacionado con el sistema se hace referencia a todas aquellas personas que se encuentran de una forma u otra vinculada con el sistema, ya sean las personas que interactúan en el proceso de desarrollo, así como las que interactúan con el sistema en sí.

**Tabla 3 Personas relacionadas con el sistema.**

| Personal Relacionado con el Sistema. | Justificación  |
|--------------------------------------|--|
| Desarrollador                        | Es la persona encargada de implementar todas las características definidas por el cliente para lograr el sistema.(Alain Vila |

|                |   |
|----------------|---|
|                | Acosta)   |
| Rol de Usuario | Son todas aquellas personas que interactúan con los contenidos generados por el desarrollador.(GEYSED, SIG's)                         |
| Cliente        | Personas o instituciones encargadas de solicitar la construcción del sistema (Centro GEYSED)  |
| Probador       | Es la persona encargada de realizar las pruebas pertinentes para asegurar el correcto funcionamiento del sistema. (Alain Vila Acosta) |

### 3.4 Fase I: Exploración

La metodología XP en su primera la fase Exploración, define el alcance general del sistema ya que los clientes describen las Historias de Usuarios. Esta fase permite que el equipo de desarrollo se familiarice con las tecnologías y herramientas que se utilizarán para el desarrollo del sistema. Se realizan estimaciones que se consideran como estimaciones primarias ya que las mismas podrían variar al analizar con más detalles cada iteración. Esta fase demora pocas semanas o pocos meses, dependiendo de la experiencia de los programadores con las tecnologías de desarrollo. **(Joskowicz, 2002)**

#### Historias de Usuario

Las Historias de Usuarios (HU) son escritas por los clientes, aunque los desarrolladores pueden brindar su ayuda en la identificación de las mismas. Las descripciones de las HU son cortas y escritas en lenguaje del usuario. Se utilizan en la parte de prueba para verificar que el sistema cumpla con las necesidades de los usuarios. El tiempo ideal para el desarrollo de las historias de usuarios es una y tres semanas y para esto cuando llega el momento de llevar a cabo la implementación de las mismas se reúnen el cliente y el equipo de desarrollo para concretar y detallar lo que tiene que hacer dicha historia de usuario.

A continuación como resultado de la Fase de exploración se expone la historia de usuario identificada para lograr el desarrollo del sistema.

**Tabla 4 HU Búsqueda en la base de datos.**

| <b>Historias de Usuarios</b>   |  |
|--|--|
| <b>Número:</b> 1   | <b>Nombre:</b> Búsqueda en la base de datos. |
| <b>Usuario:</b> Cliente  |  |
| <b>Prioridad en Negocio:</b> Alta  | <b>Riesgo en Desarrollo:</b> Alto            |
| <b>Puntos de Estimación:</b> 3   | <b>Iteración Asignada:</b> 1                 |
| <b>Descripción:</b> Obtener todas las ocurrencias encontradas en la base de datos a partir de un criterio de búsqueda.   |  |
| <b>Observaciones:</b> La funcionalidad debe lograr la revisión de todas las tablas de la base de datos, de esta forma se podrá asegurar que retornen todas las posibles respuestas relacionadas con el criterio de búsqueda definido por el usuario. Se debe también lograr una optimización en el tiempo de búsqueda. |  |

### 3.5 Fase II: Planificación

En la Fase de Planificación, el cliente define la prioridad de cada Historia de usuario que el mismo haya descrito en la fase anterior, luego de esto los programadores realizan una estimación del esfuerzo necesario para cada Historia de Usuario. Se determina un cronograma en conjunto con los clientes y programadores, según los acuerdos tomados.

El equipo de programadores mantienen un registro de la “velocidad de desarrollo”, en la que llevan un control de la suma de todos los puntos correspondiente a las HU que fueron determinadas en la última iteración. Este registro de velocidad es utilizada para determinar el tiempo que se llevará implementar un conjunto de HU.

La velocidad del proyecto es de gran utilidad para establecer la cantidad de HU que se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementarlas.

### Estimación de esfuerzos por HU

La estimación de esfuerzo se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario.

Para el logro del sistema se ha realizado la estimación de esfuerzo de la HU determinada:

**Tabla 5 Esfuerzo por historias de usuario**

| <b>Historias de Usuarios</b>  | <b>Puntos de estimación</b> |
|-------------------------------|-----------------------------|
| Búsqueda en la base de datos. | 3                           |

### **Plan de Iteraciones**

La Fase de Planificación, contiene las iteraciones del sistema antes de que el mismo sea entregado. Al estar descritas las Historias de Usuarios y los puntos de estimación de cada una de ellas, se procede a definir el Plan de iteraciones, que no es más que las partes en las que se dividirá la implementación. Hay que tener presente que el cliente es quien decide en todo momento las HU que serán implementadas en cada iteración. Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente tres semanas de duración. **(Escribano, 2002)**

Se determina realizar el sistema en una iteración que una posee una duración de tres semanas, la cual queda descrita a continuación:

#### **Iteración uno:**

En esta iteración tiene como objetivo darle cumplimiento a la HU # 1 que se define con el nombre de **Búsqueda en la base de datos**, quedando implementada dicha HU, según la descripción realizada por el cliente. Igualmente se tendrá la primera versión de pruebas.

### **Plan de duración de iteraciones**

Para lograr el desarrollo de un proyecto, la metodología de desarrollo XP, describe un ciclo de vida, en el cual se crea el plan de duración para cada una de las iteraciones que se llevan a cabo para el desarrollo del sistema. Este plan tiene como objetivo fundamental mostrar el orden en que las Historias

de usuarios han de ser implementadas, así como la duración de cada iteración definida y descrita anteriormente.

A continuación se muestra mediante una tabla el Plan de duración de las iteraciones:

**Tabla 6 Plan de duración de las iteraciones**

| <b>Iteraciones</b> | <b>Orden de las Historias de Usuarios a implementar</b> | <b>Duración total de las iteraciones</b> |
|--------------------|---|--|
| Iteración 1        | 1. Búsqueda en la base de datos.                        | 3 semanas                                |

### **3.6 Conclusiones parciales**

Con la correcta aplicación de la metodología a utilizar se comienza a desarrollar un plan de desarrollo del sistema. Primero se define todo aquel personal que intervendrá de alguna forma en el desarrollo del sistema. XP permite definir a través de las HU cuáles serán las tareas a realizar por los desarrolladores para la obtención del producto final. Dichas tareas se dividen en iteraciones definidas en este capítulo y así lograr que fluya el trabajo de forma organizada. De esta forma se da paso a la construcción de la solución para luego validar los resultados que se obtendrán mediante las pruebas pertinentes.



## Capítulo 4: Construcción de la solución propuesta

La metodología XP permite la realización del diseño del sistema, para esto es necesaria la utilización de las tarjetas C.R.C (Contenido, Responsabilidad, Colaboración) que son una técnica para lograr entender el diseño del sistema. Además XP plantea que la implementación del sistema debe realizarse de forma iterativa, logrando obtener al finalizar cada iteración un producto final, el cual es mostrado al cliente y aprobado por el mismo.

En el presente capítulo se detallan las cuatros iteraciones que se han de llevar a cabo durante la etapa de construcción del sistema. Además quedan plasmadas las tarjetas C.R.C y las tareas de ingeniería, las cuales constituyen pasos indispensables a seguir para la construcción del sistema.

### 4.1 Diseño de la solución propuesta

#### Estructura del sistema.

Para lograr un mejor entendimiento del sistema se definió la estructura del bundle (pequeña aplicación de Symfony2) con que cuenta el sistema y los paquetes que poseen el bundle.

El módulo de **BuscadorBundle**, se define la siguiente estructura: [Anexo1](#)

Con una descripción detallada de los paquetes cualquier desarrollador con un conocimiento del lenguaje de PHP puede asumir el mantenimiento y seguimiento del sistema. Este módulo cuenta con seis paquetes creados por defecto al generarlo utilizando el *framework* Symfony2.

- El paquete **Controller** posee todas aquellas clases encargadas del control y funcionamiento del servicio de búsqueda. Aquí está implementado el algoritmo para la búsqueda en la base de datos.
- El paquete **Almacen** almacena todos aquellos ficheros que sirven de sustento para realizar la búsqueda en la base de datos.
- El paquete **DependencyInjection** donde se almacenan las configuraciones del módulo.
- El paquete **Form** almacena las clases que generan los formularios del bundle.
- El paquete **Resources** en donde se almacenan las vistas del sistema así como las rutas a cada una de las páginas.

- El paquete **Servicio** es donde se almacenarán las configuraciones pertinentes para el funcionamiento del servicio de búsqueda.
- El paquete **Test** es donde se guardarán los archivos generados de las pruebas al sistema.

#### 4.1.1 Tarjetas CRC

Las Tarjetas CRC son una técnica para la representación de los sistemas, constituyen un puente de comunicación entre los diferentes participantes. **(Benk, 1999)**

Las Tarjetas CRC se dividen en tres secciones que contienen del nombre de la clase, sus responsabilidades y colaboradores.

- Nombre de las clases: Se pone el nombre de la clase a la que se está describiendo.
- Responsabilidades: Funciones que contiene implementada la clase.
- Colaboradores: Son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

A continuación se muestran las tarjetas CRC correspondientes al desarrollo del sistema:

**Tabla 7 Tarjeta CRC MotorBusquedaController**

| <b>MotorBusquedaController</b>    |                      |
|-----------------------------------|----------------------|
| <b>Responsabilidades</b>          | <b>Colaboradores</b> |
| abrirFichero                      | CConexionController  |
| cerrarFichero                     | CResultadoController |
| guardarFichero                    |                      |
| guardarFicheroNuevo               |                      |
| leerFichero                       |                      |
| servicioMotorBusqueda             |                      |
| gestionarBusqueda                 |                      |
| devolverDatosEncontradosEnFichero |                      |
| devolverDatosGuardandoEnFichero   |                      |
| realizarBusqueda                  |                      |

**Tabla 8 Tarjeta CRC CConexionController**

| <b>CConexionController</b>   |                      |
|--|----------------------|
| <b>Responsabilidades</b>   | <b>Colaboradores</b> |
| getDBName<br>abrirConexion<br>enviarConsulta<br>convConsultaObjeto<br>cerrarConexion |                      |

**Tabla 9 Tarjeta CRC ClassResultadoController**

| <b>ClassResultadoController</b>   |                      |
|---|----------------------|
| <b>Responsabilidades</b>  | <b>Colaboradores</b> |
| getValorBusqueda<br>getNombreTabla<br>getValor1<br>getValor2<br>getValor3 |                      |

#### **4.2 Fase III: Desarrollo de las Iteraciones.**

En la Fase de Planificación de la metodología XP se detallaron cada una de las HU correspondientes a cada iteración según la selección del cliente. Durante el desarrollo de las iteraciones se va realizando una revisión del plan de iteraciones ya que el mismo puede o no ser modificado. Como parte de este plan se crean tareas para ayudar a organizar la implementación exitosa de la HU.

Estas tareas en la que se descomponen las HU son las llamadas tareas de ingeniería, que son asignadas a una persona perteneciente al equipo de desarrollo, que el mismo sea quien responda por la implementación asignada. Estas tareas son escritas por los programados y no por el cliente, ya que son para uso estricto de los programadores.

La planificación que se lleva a cabo para el desarrollo del sistema, está compuesta por cuatro iteraciones, permitiendo que al final de la última iteración se logre un producto con todas las restricciones y características deseadas por el cliente.

## 4.2 Tareas de la Ingeniería

Para llevar a cabo las tareas de la ingeniería se cuenta con una tabla que permite definir cada una de las actividades en las que estará asociada cada historia de usuario. A continuación se detallan las tareas asignadas por cada una de las iteraciones.

### 4.2.1 Iteración uno

En esta iteración se implementara la HU número uno con el objetivo de realizar una búsqueda en una base de datos en toda su extensión así como una vez hecha la búsqueda tratar de minimizar lo más posible el tiempo que demore esta en hacerse.

**Tabla 10 HU a realizar y tiempo de duración.**

| Historias de Usuarios               | Estimación | Real      |
|-------------------------------------|------------|-----------|
| HU#1: Búsqueda en la base de datos. | 3 semanas  | 3 semanas |

A continuación se muestra mediante tablas las tareas de ingeniería en la que la HU mencionada anteriormente fue desglosada, para lograr un mejor funcionamiento del sistema.

**Tabla 11 Tarea de Ingeniería #1, Iteración 1.**

| Tareas de la Ingeniería  |                             |
|--|-----------------------------|
| <b>No. De la tarea:</b> 1  | <b>No. De la HU:</b> 1      |
| <b>Nombre de la tarea:</b> Recorrer toda la base de datos.   |                             |
| <b>Tipo de tarea:</b> Desarrollo   | <b>Puntos estimados:</b> 3  |
| <b>Fecha inicio:</b> 05/12/2011  | <b>Fecha fin:</b> 8/12/2011 |
| <b>Programador responsable:</b> Alain Vila Acosta  |                             |
| <b>Descripción:</b> Esta tarea consiste en lograr obtener cada uno de los esquemas que posea la base de datos así como las tablas que pertenecen a cada uno de estos esquemas. |                             |

**Tabla 12 Tarea de Ingeniería #2, Iteración 1.**

| <b>Tareas de la Ingeniería</b>  |                              |
|---|------------------------------|
| <b>No. De la tarea:</b> 2   | <b>No. De la HU:</b> 1       |
| <b>Nombre de la tarea:</b> Verificar campo de tabla.  |                              |
| <b>Tipo de tarea:</b> Desarrollo  | <b>Puntos estimados:</b> 3   |
| <b>Fecha inicio:</b> 9/12/2011  | <b>Fecha fin:</b> 12/12/2011 |
| <b>Programador responsable:</b> Alain Vila Acosta   |                              |
| <b>Descripción:</b> Esta tarea consiste en obtener o verificar la existencia del campo <b>name</b> o <b>nombre</b> existente en cada tabla para así poder comparar con el criterio de búsqueda. |                              |

**Tabla 13 Tarea de Ingeniería #3, Iteración 1.**

| <b>Tareas de la Ingeniería</b>  |                              |
|---|------------------------------|
| <b>No. De la tarea:</b> 3   | <b>No. De la HU:</b> 1       |
| <b>Nombre de la tarea:</b> Construir consulta final.  |                              |
| <b>Tipo de tarea:</b> Desarrollo  | <b>Puntos estimados:</b> 3   |
| <b>Fecha inicio:</b> 15/12/2011   | <b>Fecha fin:</b> 17/12/2011 |
| <b>Programador responsable:</b> Alain Vila Acosta   |                              |
| <b>Descripción:</b> Esta tarea consiste en al ir obteniendo los nombres de los esquemas con sus tablas así como el campo <b>name</b> o <b>nombre</b> construir la consulta final que devolverá los datos solicitados. |                              |

**Tabla 14 Tarea de Ingeniería #4, Iteración 1.**

| <b>Tareas de la Ingeniería</b>                    |                              |
|---|------------------------------|
| <b>No. De la tarea:</b> 4                         | <b>No. De la HU:</b> 1       |
| <b>Nombre de la tarea:</b> Resultado Final.       |                              |
| <b>Tipo de tarea:</b> Desarrollo                  | <b>Puntos estimados:</b> 3   |
| <b>Fecha inicio:</b> 17/12/2011                   | <b>Fecha fin:</b> 18/12/2011 |
| <b>Programador responsable:</b> Alain Vila Acosta |                              |

|  |
|--|
| <b>Descripción:</b> Esta tarea consiste en luego de haber recorrido toda la base de datos mostrar todos los datos en un listado con un correcto formato. |
|--|

**Tabla 15 Tarea de Ingeniería #5, Iteración 1.**

| <b>Tareas de la Ingeniería</b>  |                              |
|---|------------------------------|
| <b>No. De la tarea:</b> 5   | <b>No. De la HU:</b> 1       |
| <b>Nombre de la tarea:</b> Reducir tiempo de búsqueda.  |                              |
| <b>Tipo de tarea:</b> Desarrollo  | <b>Puntos estimados:</b> 3   |
| <b>Fecha inicio:</b> 20/12/2011   | <b>Fecha fin:</b> 26/12/2011 |
| <b>Programador responsable:</b> Alain Vila Acosta   |                              |
| <b>Descripción:</b> Esta tarea consiste en optimizar mediante el uso de ficheros donde se almacenarán diferentes datos el tiempo búsqueda en la base de datos y así minimizar el tiempo de espera de la búsqueda. |                              |

### 4.3 Pruebas

Una de las características fundamentales de la metodología XP es su marcado énfasis en la realización de pruebas, comprobando así el funcionamiento del sistema. Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo a medida que se realiza el sistema. Por esta razón se debe definir en el proceso de la ingeniería del software (**Carmen, 2008**). Las pruebas ayudan a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.

La metodología XP divide las pruebas en dos grupos: las pruebas unitarias, las cuales la desarrollan los programadores, estas son las encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (**Pressman**).

#### 4.3.1 Pruebas Unitarias

Las pruebas unitarias se utilizan para verificar el funcionamiento del sistema. Una prueba unitaria es la verificación de un módulo determinado dentro de un sistema. Estas pruebas son llevadas a cabo por

los programadores de cada módulo. Las pruebas unitarias poseen gran beneficio ya que mediante ellas, se puede asegurar que un determinado módulo cumpla con el funcionamiento esperado.

Las pruebas unitarias se consideran una actividad fundamental de XP ya que brindan una visión de lo que se quiere realizar. Además demuestra que lo implementado es lo que se pensaba desde el principio.

A continuación se exponen algunas utilidades de este tipo de pruebas:

- Ayudan a la producción de un código de mayor calidad.
- Permite la detección rápida de errores cuando se programan nuevas funcionalidades o se realizan cambios en el código.
- Sirven como pequeña fuente de documentación sobre qué es lo que se espera que haga el código.
- Realizar pruebas unitarias obliga al programador a escribir el código en pequeñas porciones.

Para el desarrollo de las pruebas se utilizó la herramienta SimpleTest. SimpleTest es una herramienta para realizar pruebas unitarias al código PHP. Consta de una carpeta en donde se almacena todas las clases encargadas de realizarle las pruebas al código. Dicha carpeta se coloca dentro del proyecto que se va a probar o en algún directorio global de PHP. Una vez instalado el software se definen los casos de pruebas como se puede observar en el [Anexo 2](#).

El incluir el fichero autorun.php posibilita poder extender la clase UnitTestCase, permitiendo ejecutar las pruebas. Además el autorun.php se encarga de correr las pruebas sin necesidad de que se tenga que instanciar nada y también es suficientemente inteligente para darse cuenta de estar en la consola o directamente en el navegador. La función setUp se ejecuta automáticamente antes de correr cualquier prueba. Finalmente SimpleTest corre todas las pruebas que comienzan con el nombre test en este caso existen 4 de ellas:

**testServicio:** Prueba que el resultado de consumir el servicio Web devuelva un objeto que contiene arreglos.

**Resultado esperado:**

El servicio web devuelve un objeto de arreglos

```
object(stdClass)[16]
  public 'Struct' =>
    array
      0 =>
        object(stdClass)[17]
          public 'valorBusqueda' => string 'Docente 4' (length=9)
          public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
          public 'valor1' => string 'El Docente 4 se inaugurÃ³ en el aÃ±o 2005
Como dato de interÃ©s, allÃ­ se encuentra la Cadeca # 2 de nuestra universidad
          public 'valor2' => string '837-2508, 837-2508' (length=18)
          public 'valor3' => string '' (length=0)
      1 =>
        object(stdClass)[18]
          public 'valorBusqueda' => string 'Cajero del Docente 4' (length=20)
          public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
          public 'valor1' => string 'El Cajero AutomÃ¡tico del Docente 4 entrÃ
El suministro de dinero al cajero es realizado por el banco de la universidad
          public 'valor2' => string '835-8400, 835-8398' (length=18)
          public 'valor3' => string '' (length=0)
```



**testFuncionGestionarBusqueda:** Prueba que el resultado de la función remota del servicio Web devuelva un arreglo de objetos.

**Resultado esperado:**

La funcion gestionarBusqueda devuelve arreglo

```
array
0 =>
  object (CResultadoController)[30]
    public 'valorBusqueda' => string 'Restaurante "La Casona"' (length=23)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El restaurante "La Casona", se encuentra ubicado
    public 'valor2' => string '835-8079' (length=8)
    public 'valor3' => null
1 =>
  object (CResultadoController)[27]
    public 'valorBusqueda' => string 'Restaurante "El RanchÃ³n"' (length=25)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El Restaurante "El RanchÃ³n" se crea en la UCI c
TodavÃa no estÃa concluido.
' (length=260)
  public 'valor2' => string 'No disponible' (length=13)
  public 'valor3' => null
2 =>
  object (CResultadoController)[32]
    public 'valorBusqueda' => string 'Restaurante "El FramboyÃn"' (length=27)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El Restaurante "El FramboyÃn" presta servicios
Tiene 20 trabajadores distribuidos en 12 horas por turnos.
' (length=250)
  public 'valor2' => string '6454136' (length=7)
  public 'valor3' => null
3 =>
  object (CResultadoController)[29]
    public 'valorBusqueda' => string 'Restaurantes' (length=12)
    public 'nombreTabla' => string 'tb_categorias' (length=13)
    public 'valor1' => null
    public 'valor2' => null
    public 'valor3' => null
```

**testFuncionDevolverDatosEncontradosEnFichero:** Prueba que el resultado de la función encargada de buscar valores en el fichero devuelva un arreglo de objetos.

**Resultado esperado:**

La función `devolverDatosEncontradosEnFichero` devuelve arreglo

```
array
  0 =>
    object(CResultadoController)[16]
      public 'valorBusqueda' => string 'Docente 4' (length=9)
      public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
      public 'valor1' => string 'El Docente 4 se inauguró en el año 2005, el
Como dato de interés, allí se encuentra la Cadeca # 2 de nuestra universidad.
      public 'valor2' => string '837-2508, 837-2508' (length=18)
      public 'valor3' => null
  1 =>
    object(CResultadoController)[30]
      public 'valorBusqueda' => string 'Cajero del Docente 4' (length=20)
      public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
      public 'valor1' => string 'El Cajero Automático del Docente 4 entró en
El suministro de dinero al cajero es realizado por el banco de la universidad.
      public 'valor2' => string '835-8400, 835-8398' (length=18)
      public 'valor3' => null
  2 =>
    object(CResultadoController)[33]
      public 'valorBusqueda' => string 'Consultorio Médico (Edificio 49)' (len
      public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
      public 'valor1' => string 'En este consultorio se atienden a los estudian
Consulta Facultad 3.
      Doctor: Jorge Radis García Rondán
      Enfermera: Vivian Rodríguez
Consulta Facultad 4.
      ... (length=645)
      public 'valor2' => string '837-2037' (length=8)
      public 'valor3' => null
```

**testFuncionDevolverDatosGuardandoEnFichero:** Prueba que el resultado de la función que guarda los valores encontrados en un fichero devuelva un arreglo.

**Resultado esperado:**

La función `devolverDatosGuardandoEnFichero` devuelve arreglo

```
array
0 =>
  object(CResultadoController)[37]
    public 'valorBusqueda' => string 'Docente 4' (length=9)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El Docente 4 se inauguró en el año 2005, el
Como dato de interés, allí se encuentra la Cadeca # 2 de nuestra universidad.
    public 'valor2' => string '837-2508, 837-2508' (length=18)
    public 'valor3' => null
1 =>
  object(CResultadoController)[38]
    public 'valorBusqueda' => string 'Docente 3' (length=9)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El Docente 1, conocido como el "Docente Chino"
' (length=409)
    public 'valor2' => string '837-2519, 837-2519' (length=18)
    public 'valor3' => null
2 =>
  object(CResultadoController)[39]
    public 'valorBusqueda' => string 'Docente 5' (length=9)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El Docente 5, conocido como el "Docente Azul"
Entre sus oficinas principales están las de Postgrado y Capital Humano y tien
' (length=508)
    public 'valor2' => string '837-2470, 837-2463' (length=18)
    public 'valor3' => null
3 =>
  object(CResultadoController)[40]
    public 'valorBusqueda' => string 'Docente 6' (length=9)
    public 'nombreTabla' => string 'tb_socioeconomica' (length=17)
    public 'valor1' => string 'El Docente 6 todavía está en manos del grupo
```

Cualquier otro resultado nos arrojaría errores. Estas pruebas ejecutadas en el navegador se verían como se muestra en el [Anexo 3](#), pero también se pueden ejecutar en consola sin ningún problema.

### 4.3.2 Pruebas de Aceptación

Las pruebas de aceptación no son más que validaciones que se le realiza al sistema para ver si cumple con el funcionamiento esperado. Estas pruebas son definidas por el usuario del sistema y las prepara el equipo de trabajo.

Las pruebas de aceptación son pruebas de caja negra que se realizan partiendo de las Historias de Usuario, estas pueden llevarse a cabo varias veces hasta que el sistema cumpla con la funcionalidad deseada. Los propios clientes son los responsables de que los resultados de estas pruebas sean correctos. En caso de que fallen varias pruebas deben de indicar el orden en que las mismas han de ser resueltas. Una Historia de Usuario no puede indicarse resuelta hasta que la misma no pase correctamente por las pruebas de aceptación.

El objetivo específico de esta prueba es garantizar que los requerimientos han sido cumplidos y que el sistema ha sido aceptado.

A continuación se muestran las pruebas de aceptación para cada Historia de Usuario:

**Tabla 16 Caso de Prueba de Aceptación HU1\_P1**

| <b>Caso de Prueba de Aceptación</b>   |              |
|---|--------------|
| <b>Código:</b> HU1_P1   | <b>HU:</b> 1 |
| <b>Nombre de la prueba:</b> Comprobar búsqueda en la base de datos  |              |
| <b>Descripción:</b> A través de esta prueba se verifica la correcta obtención de los resultados deseados.   |              |
| <b>Condiciones de ejecución:</b><br>El módulo de búsqueda debe estar conectado a una base datos, dicha base de datos será SIG_UCI_New posee 3 esquemas y un total de 26 tablas. |              |

- [-] SIG\_UCI\_New
  - + Catálogos (2)
  - [-] Esquemas (3)
    - + public
    - + rutas
    - + vb
  - Replicación (0)


### Entradas/ Pasos de ejecución:

Se entra un criterio de búsqueda el cual puede ser cualquier palabra.



---

Luego de ejecutar la búsqueda se muestra el siguiente resultado.



---

[Docente 4](#)

**Lugar:**  
**Telefonos:** 837-2508, 837-2508

**Descripcion:** El Docente 4 se inauguró en el año 2005, el mismo posee 20 laboratorios, 5 salones, 20 aulas y 5 departamentos, los cuales son: Ciencias Básicas, Humanidades, Técnicas de Programación, Sistemas Digitales e Ingeniería y Sistema de Gestión de Software. Además en él radican 2 facultades: la Facultad 6 y la Facultad 9. Como dato de interés, allí se encuentra la Cadeca # 2 de nuestra universidad.

[Docente 3](#)

**Lugar:**  
**Telefonos:** 837-2519, 837-2519



**Descripcion:** El Docente 1, conocido como el "Docente Chino" se inauguró en Enero de 2005 por el Presidente de la República Popular China Hu Jintao, el mismo posee 20 laboratorios, 5 salones, 20 aulas y 5 departamentos, los cuales son: Ciencias Básicas, Humanidades, Técnicas de Programación, Sistemas Digitales e Ingeniería y Gestión de Software. Además en él radican 2 facultades: la Facultad 10 y la Facultad 11.


**Resultados esperados:**

Se muestra todas las ocurrencias encontradas en la base de datos.








**Evaluación de la prueba:** Prueba satisfactoria.

Tabla 17 Caso de Prueba de Aceptación HU1\_P2

| Caso de Prueba de Aceptación  |              |
|---|--------------|
| <b>Código:</b> HU1_P2   | <b>HU:</b> 1 |
| <b>Nombre de la prueba:</b> Comprobar búsqueda en la base de datos  |              |
| <b>Descripción:</b> A través de esta prueba se verifica la correcta obtención de los resultados deseados.   |              |
| <b>Condiciones de ejecución:</b><br>El módulo de búsqueda debe estar conectado a una base datos, dicha base de datos será SIG_UCI_New posee 3 esquemas y un total de 26 tablas.<br> |              |
| <b>Entradas/ Pasos de ejecución:</b><br>Se entra un criterio de búsqueda el cual puede ser cualquier palabra. En este caso se entra una palabra errónea.<br>                      |              |
| Luego de ejecutar la búsqueda se muestra el siguiente resultado.  |              |

|  |
|--|
|  <b>Buscador</b>          |
| <input type="text" value="#\$@ddd"/> <input type="button" value="Buscar"/>                                 |
| <p>No se ha encontrado ningun resultado</p>  |
| <p><b>Resultados esperados:</b><br/>Se muestra un mensaje indicando que no se ha encontrado resultado.</p> |
| <p><b>Evaluación de la prueba:</b> Prueba satisfactoria.</p>   |

**Tabla 18 Caso de Prueba de Aceptación HU1\_P3**

| Caso de Prueba de Aceptación  |              |
|---|--------------|
| <b>Código:</b> HU1_P3   | <b>HU:</b> 1 |
| <b>Nombre de la prueba:</b> Comprobar búsqueda en la base de datos  |              |
| <b>Descripción:</b> A través de esta prueba se verifica la correcta obtención de los resultados deseados.   |              |
| <p><b>Condiciones de ejecución:</b><br/>El módulo de búsqueda debe estar conectado a una base datos, dicha base de datos será SIG_UCI_New posee 3 esquemas y un total de 26 tablas.</p> <ul style="list-style-type: none"> <li>[-]  SIG_UCI_New <ul style="list-style-type: none"> <li>[+]  Catálogos (2)</li> <li>[-]  <b>Esquemas (3)</b> <ul style="list-style-type: none"> <li>[+]  public</li> <li>[+]  rutas</li> <li>[+]  vb</li> </ul> </li> <li> Replicación (0)</li> </ul> </li> </ul> |              |
| <p><b>Entradas/ Pasos de ejecución:</b><br/>Se entra un criterio de búsqueda el cual puede ser cualquier palabra. En este caso se ejecuta la búsqueda sin haber entrado ningún criterio.</p>  |              |

Buscador

Luego de ejecutar la búsqueda se muestra el siguiente resultado.

Buscador

Debe entrar un criterio de búsqueda

**Resultados esperados:**  
Se muestra un mensaje indicando que debe entrar algún criterio de búsqueda.

**Evaluación de la prueba:** Prueba satisfactoria.

Tabla 19 Caso de Prueba de Aceptación HU1\_P4

| Caso de Prueba de Aceptación  |              |
|---|--------------|
| <b>Código:</b> HU1_P4   | <b>HU:</b> 1 |
| <b>Nombre de la prueba:</b> Comprobar búsqueda en la base de datos  |              |
| <b>Descripción:</b> A través de esta prueba se verifica la correcta obtención de los resultados deseados.   |              |
| <b>Condiciones de ejecución:</b><br>El módulo de búsqueda debe estar conectado a una base de datos, dicha base de datos será DATOSSIG posee 12 esquemas y un total de 243 tablas. |              |



- [-] DATOSSIG
  - [+] Catálogos (2)
  - [-] Esquemas (12)
    - [+] Poblacion
    - [+] cartogramas
    - [+] direcciones
    - [+] mod\_estructuracomp
    - [+] mod\_miscelaneas
    - [+] mtd\_100000
    - [+] mtd\_250000
    - [+] mtd\_500000
    - [+] mtd\_ciudades
    - [+] public
    - [+] rutas
    - [+] temporal
    - Replicación (0)

**Entradas/ Pasos de ejecución:**

Se entra un criterio de búsqueda el cual puede ser cualquier palabra.

**Buscador**

---

Luego de ejecutar la búsqueda se muestra el siguiente resultado.

**Buscador**

---

[iddpa](#)

**Lugar:** nom\_campoestruc  
**Telefonos:**  
**Descripcion:**

[nom organo](#)

















**Lugar:** nom\_campoestruc  
**Telefonos:**  
**Descripcion:**



**Resultados esperados:**

Se muestra todas las ocurrencias encontradas en la base de datos.

**Evaluación de la prueba:** Prueba satisfactoria.












**Tabla 20 Caso de Prueba de Aceptación HU1\_P5**

| <b>Caso de Prueba de Aceptación</b>  |              |
|--|--------------|
| <b>Código:</b> HU1_P5  | <b>HU:</b> 1 |
| <b>Nombre de la prueba:</b> Comprobar búsqueda en la base de datos   |              |
| <b>Descripción:</b> A través de esta prueba se verifica la correcta obtención de los resultados deseados.  |              |
| <p><b>Condiciones de ejecución:</b><br/>           El módulo de búsqueda debe estar conectado a una base datos, dicha base de datos será DATOSSIG posee 12 esquemas y un total de 243 tablas.</p> <ul style="list-style-type: none"> <li>[-]  DATOSSIG               <ul style="list-style-type: none"> <li>[+]  Catálogos (2)</li> <li>[-]  Esquemas (12)                   <ul style="list-style-type: none"> <li>[+]  Poblacion</li> <li>[+]  cartogramas</li> <li>[+]  direcciones</li> <li>[+]  mod_estructuracomp</li> <li>[+]  mod_miscelaneas</li> <li>[+]  mtd_100000</li> <li>[+]  mtd_250000</li> <li>[+]  mtd_500000</li> <li>[+]  mtd_ciudades</li> <li>[+]  public</li> <li>[+]  rutas</li> <li>[+]  temporal</li> <li> Replicación (0)</li> </ul> </li> </ul> </li> </ul> |              |
| <p><b>Entradas/ Pasos de ejecución:</b><br/>           Se entra un criterio de búsqueda el cual puede ser cualquier palabra. En este caso se entra una palabra errónea.</p>  |              |

|   |
|---|
|  <b>Buscador</b>           |
| <input type="text" value="avafvfs"/> <input type="button" value="Buscar"/>                                  |
| <p>Luego de ejecutar la búsqueda se muestra el siguiente resultado.</p>                                     |
|  <b>Buscador</b>           |
| <input type="text" value="avafvfs"/> <input type="button" value="Buscar"/>                                  |
| <p><b>No se ha encontrado ningun resultado</b></p>  |
| <p><b>Resultados esperados:</b><br/>Se muestra mensaje indicando que no se encontrado ningún resultado.</p> |
| <p><b>Evaluación de la prueba:</b> Prueba satisfactoria.</p>  |

**Tabla 21 Caso de Prueba de Aceptación HU1\_P5**

| <b>Caso de Prueba de Aceptación</b>  |              |
|--|--------------|
| <b>Código:</b> HU1_P6  | <b>HU:</b> 1 |
| <b>Nombre de la prueba:</b> Comprobar búsqueda en la base de datos   |              |
| <b>Descripción:</b> A través de esta prueba se verifica la correcta obtención de los resultados deseados.  |              |
| <b>Condiciones de ejecución:</b><br>El módulo de búsqueda debe estar conectado a una base datos, dicha base de datos será DATOSSIG posee 12 esquemas y un total de 243 tablas. |              |

- [-]  DATOSSIG
  - [+]  Catálogos (2)
  - [-]  Esquemas (12)
    - [+]  Poblacion
    - [+]  cartogramas
    - [+]  direcciones
    - [+]  mod\_estructuracomp
    - [+]  mod\_misclaneas
    - [+]  mtd\_100000
    - [+]  mtd\_250000
    - [+]  mtd\_500000
    - [+]  mtd\_ciudades
    - [+]  public
    - [+]  rutas
    - [+]  temporal
    -  Replicación (0)

**Entradas/ Pasos de ejecución:**

Se entra un criterio de búsqueda el cual puede ser cualquier palabra. En este caso se ejecuta la búsqueda sin ningún criterio de búsqueda.

 **Buscador**

---

Luego de ejecutar la búsqueda se muestra el siguiente resultado.

 **Buscador**

---

**Debe entrar un criterio de busqueda**

**Resultados esperados:**

Se muestra mensaje indicando que debe entrar un criterio de búsqueda.

**Evaluación de la prueba:** Prueba satisfactoria.

Se utilizan también, como planillas para realizar las pruebas, las determinadas por la UCI que se encuentran anexas a este documento.

#### **4.4 Conclusiones Parciales**

En este capítulo se documentó todo el proceso de construcción de la solución propuesta hasta concluir con las pruebas de la misma. Con ayuda de la metodología seleccionada para su desarrollo permitió en primer lugar dar una descripción del sistema a través de las tarjetas CRC para un mejor entendimiento del mismo, segundo, se dividió el trabajo en iteraciones con el objetivo de agrupar las tareas según un orden lógico y así poder mostrar resultados periódicamente. Asimismo permitió la fluidez a la hora de intercambiar ideas con el cliente sobre el progreso y aceptación del trabajo realizado.

## Conclusiones

Luego de finalizado el trabajo se arribaron a las siguientes conclusiones:

- ✓ Se desarrolló un servicio Web que permite la búsqueda de objetivos geográficos en bases de datos espaciales compuesto por un módulo el cual hace función de motor de búsqueda para la base de dato que le sea configurado.
- ✓ El sistema fue desarrollado haciendo uso de herramientas libres y cumpliendo así con los estándares especificados por el centro **GEYSED** y la Universidad de las Ciencias Informáticas (**UCI**).
- ✓ Integrar el modulo con PLASMA permite el aporte de un servicio más para dicha plataforma además de aumentar la usabilidad de la misma.
- ✓ El servicio permite realizar búsquedas en bases de datos implementadas en servidores PostgreSQL de cualquier temática convirtiéndose en una herramienta versátil y útil.
- ✓ El servicio tiene la capacidad de realizar búsquedas en toda la base de datos a la cual esté conectado permitiendo así obtener todos los resultados posibles según el criterio de búsqueda.

## Recomendaciones

Para la realización de futuras mejoras en el servicio Web de búsqueda se recomienda:

- ✓ Eliminar la creación de los ficheros de consultas para mejorar la velocidad de búsqueda y hallar una solución más eficiente que realice la misma función.
- ✓ Optimizar el modulo desarrollado logrando así la búsqueda de los objetivos geográficos solamente en aquellas tablas que cuente con la información solicitada por el usuario.
- ✓ Extender el uso del servicio de búsqueda no solamente a bases de datos espaciales.

## Bibliografía

1. Beck, F.M.Y.K. Planenando en Programacion Extrema. 2000.
2. Brenes, Carlos. 2005. OIRSA [En línea] 2005. [www.oirsa.org/prepac/documentos\\_generales/sistemas\\_inf\\_geog.pdf](http://www.oirsa.org/prepac/documentos_generales/sistemas_inf_geog.pdf).
3. Cifuentes Sánchez, V.J. y González Rojas. Infraestructura de Datos Espaciales de la Confederación. Sevilla. : s.n., 2010.
4. Consortium, W. (11 de 2 de 2004). <http://www.w3c.org>. Recuperado el 27 de 11 de 2011, de <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211#whatis>
5. Carlos E. Arcancel: Web Map Services: La nueva era de información geográfica digital. Revista Ciencia e Innovación Tecnológica. Número 126. 2008.
6. ICDE. (n.d.). Infraestructura Colombiana de Datos Espaciales. Retrieved noviembre 14, 2011, from <http://www.icde.org.co/Web/guest/wiki/-/wiki/Wiki%20de%20la%20ICDE/Objetos+Geogr%C3%A1ficos;jsessionid=61CDA0FCFBA30770156890706F91C8DE>
7. <http://www.opengeospatial.org>
8. Olaya, V. (2010). Sistema de Información Geográfica
9. Tidwell, D Doug, Snell, James y Kulchenko, Pavel. 2002. Programming Web Service with SOAP. s.l. : O'Really, 2002.
10. [www.oirsa.org/prepac/documentos\\_generales/sistemas\\_inf\\_geog.pdf](http://www.oirsa.org/prepac/documentos_generales/sistemas_inf_geog.pdf).



## Referencias Bibliográficas

1. **(Andalucía, 2005)** Andalucía, Junta de. 2005. ideAndalucía (Infraestructura de Datos Espaciales de Andalucía). [Online] 2005. [Cited: 11 10, 2011.]
2. **(Autores, 2007)** Autores, Varios. 2007. PergaminoVirtual. [Online] 5 3, 2007. [Cited: 04 23, 2012.] <http://www.pergaminovirtual.com>.
3. **(Beck, 2000)** Beck, F.M.Y.K. 2000. Planenando en Programacion Extrema. 2000.
4. **(Benk, 1999)** Benk, Kent. 1999. Extreme Progrmming Explained. s.l. : Adisson-Wesley, 1999.
5. **(Brenes, 2005)** Brenes, Carlos. 2005. OIRSA. [Online] 2005.
6. **(Carmen, 2008)** Carmen. 2008. Metodologías ágiles para el desarrollo de software. [Online] 2008. [Cited: 02 2012, 16.] <http://www.willydev.net/descargas/masyxp.pdf>.
7. **(Consortium, 2004)** Consortium, W3C. 2004. [Online] 02 11, 2004. [Cited: 11 15, 2011.] <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>.
8. **(Corchete, 2011)** Corchete, Antonio Gilbert Paneque. 2011. Plataforma de Servicios de Mapas para la georreferenciación de objetivos socioeconómicos. GEYSED, Universidad de las Ciencias Informaticas. 2011. Tesis Ingenieril.
9. **(Escribano, 2002)** Escribano, Gerardo Fernández. 2002. Introduccion a la Extreme Programming. 2002.
10. **(Group, 1996)** Group, PostgreSQL Global Development. 1996. PostgreSQL. [Online] 1996. [Cited: 02 25, 2012.] <http://www.postgresql.org/developer/>.
11. **(Guting, 2004)** Guting, R. (2004). An introduction to spatial database systems. The International Journal on Very Large Data Bases
12. **(Haithcoat, 1999)** Haithcoat, T. (1999). Bases de Datos Espaciales como modelos de la realidad. Columbia: University of Missouri.
13. **(ICDE, 2011)** ICDE. (n.d.). Infraestructura Colombiana de Datos Espaciales. Retrieved noviembre 14, 2011, from <http://www.icde.org.co/>
14. **(Infa, 2004)** Infraestructura de datos Espaciales de la República de Cuba. [Online] 2004. <http://www.iderc.co.cu>.
15. **(José H. Canós)** José H. Canós, Patricio Letelier yM<sup>a</sup> Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. DSIC , Universidad Politécnica de Valencia. Valencia : s.n.
16. **(Joskowicz, 2002)** Joskowicz, José. 2002. Reglas y Prácticas en eXtreme Programming. [Online] 2002. <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.

17. **(Machuca, 2008)** Morales Machuca, C. A. (2008). Estado del Arte: Servicios Web.
18. **(Netbeans)** Netbeans. Netbeans. [Online] [Cited: 02 12, 2012.] <http://netbeans.org/community/releases/70/>.
19. **(Newkirk, 2001)** Newkirk, J., Martin R.C. 2001. "Extreme Programming in Practice". s.l. : Addison-Wesley, 2001.
20. **(Oficina, 2010)** Oficina Española del W3C. [En línea] 5 de 6 de 2010. <http://www.w3c.es/>.
21. **(Olaya, 2010)** Olaya, V. (2010). Sistema de Información Geográfica.
22. **(PHP, 2012)** PHP.net. [Online] [Cited: 02 15, 2012.] <http://www.php.net/docs.php>.
23. **(Pressman, 2005)** Pressman, R. S. Ingeniería del software.
24. **(Sherman, 2004-2007)** Sherman, Gary E. 2004-2007. Quantum GIS. [Online] 2004-2007. [Cited: 04 26, 2012.] <http://www.qgis.org>.
25. **(Rigaux, 2001)** Rigaux, P. and Scholl, M. & Voisard, A. 2001. Introduction to Spatial Databases: Applications to GIS. 2001. p. 400.
26. **(Site, 2012)** Site, Admin. Symfony. [Online] [Cited: 01 23, 2012.] <http://symfony.com/why-use-a-framework>.
27. **(Tidwell, 2002)** Tidwell, D Doug, Snell, James y Kulchenko, Pavel. 2002. Programming Web Service with SOAP. s.l. : O'Really, 2002.
28. **(Van de Putte, 2004)** Van de Putte, Geert, y otros. Using Web Services for Bussiness Integrations. IBM, 2004: 0738425486.

## Anexos

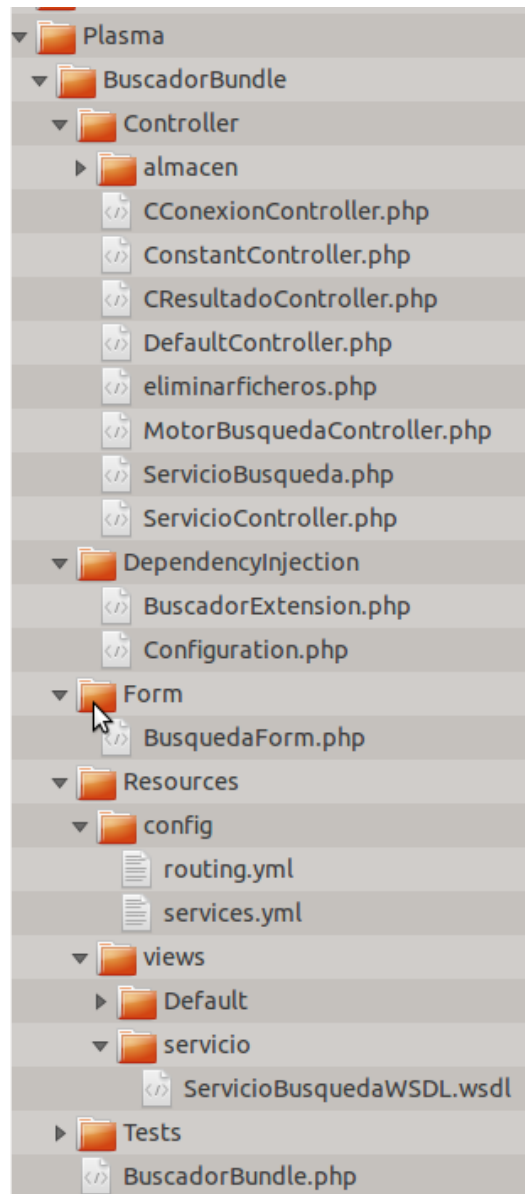


Ilustración 6 Anexo 1\_Estructura BundleBusqueda

```

<?php
require_once 'simpletest/autorun.php';
require_once 'MotorBusquedaController.php';
class AreaCasosdePrueba extends UnitTestCase {
    function setUp() {
    }
    function testServicio() {
        $client = new soapclient('http://localhost/modulo_busqueda/web/app_dev.php/servicio/ServicioBusqueda?wsdl');
        $result = $client->_call('servicioMotorBusqueda', array('DBHost' => 'localhost', 'DBPort' => '5432', 'DBName' => 'postgres'));
        $this->assertTrue(is_object($result));
        echo "El servicio web devuelve un objeto de arreglos<br>";
        echo var_dump($result);
    }
    function testFuncionGestionarBusqueda() {
        $motor = new MotorBusquedaController();
        $motor->servicioMotorBusqueda('localhost', '5432', 'SIG_UCI_SE', 'postgres', 'postgres', utf8_encode('restante'));
        $result = $motor->gestionarBusqueda(utf8_encode('restante'), 'detalles', 'telefonos', null);
        $this->assertTrue(is_array($result));
        echo "La funcion gestionarBusqueda devuelve arreglo<br>";
        echo var_dump($result);
    }
    function testFuncionDevolverDatosEncontradosEnFichero() {
        $motor = new MotorBusquedaController();
        $motor->servicioMotorBusqueda('localhost', '5432', 'SIG_UCI_SE', 'postgres', 'postgres', utf8_encode('4'));
        $result = $motor->devolverDatosEncontradosEnFichero(utf8_encode('4'), 'detalles', 'telefonos', null);
        $this->assertTrue(is_array($result));
        echo "La funcion devolverDatosEncontradosEnFichero devuelve arreglo<br>";
        echo var_dump($result);
    }
    function testFuncionDevolverDatosGuardandoEnFichero() {
        $motor = new MotorBusquedaController();
        $motor->servicioMotorBusqueda('localhost', '5432', 'SIG_UCI_SE', 'postgres', 'postgres', utf8_encode('docente'));
        $result = $motor->devolverDatosGuardandoEnFichero(utf8_encode('docente'), 'detalles', 'telefonos', null);
        $this->assertTrue(is_array($result));
        echo "La funcion devolverDatosGuardandoEnFichero devuelve arreglo<br>";
        echo var_dump($result);
    }
}
?>

```

Ilustración 7 Anexo 2\_Casos de prueba



Ilustración 8 Anexo 3\_Resultado de los casos de prueba