

*Universidad de las Ciencias Informáticas*

*Facultad 6*



**Título: Sistema de Información Geográfica para la  
Universidad de las Ciencias Informáticas sobre dispositivos  
móviles basado en los servicios Web estandarizados por la  
*Open Geospatial Consortium.***

**Trabajo para optar por el Título de Ingeniería en  
Ciencias Informáticas**

**Autor: Daryl Hernández Couce**

**Tutor: Lic. Mario Liosbel Díaz Abreu**

**Co-Tutor: Ing. Odiel Estrada Molina**

Ciudad de La Habana, Junio del 2012

“Año 54 de la Revolución”



*"Ser el hombre más rico del cementerio no me interesa... Lo que me importa es irme a la cama cada noche sabiendo que hemos*

*hecho algo maravilloso."*

*Steve Jobs (1955 - 2011).*

*Declaración de Autoría*

Yo: Daryl Hernández Couce declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del 2012.

Daryl Hernández Couce

Lic. Mario Liosbel Díaz Abreu.

\_\_\_\_\_  
FIRMA DEL AUTOR

\_\_\_\_\_  
FIRMA DEL TUTOR

## *Datos de Contacto*

### **Autor**

Nombre: Daryl.

Apellidos: Hernández Couce.

Correo electrónico: [dcouce@estudiantes.uci.cu](mailto:dcouce@estudiantes.uci.cu)

### **Tutor**

Nombre: Mario Liosbel.

Apellidos: Díaz Abreu.

Correo electrónico: [wariol@uci.cu](mailto:wariol@uci.cu)

### **Co-Tutor**

Nombre: Odiel.

Apellidos: Estrada Molina.

Correo electrónico: [oestrada@uci.cu](mailto:oestrada@uci.cu)

## *Agradecimientos*

*A mis padres y mis abuelas que siempre me han apoyado.*

*A mis padrastros Fernando y Mayling que me han querido como un hijo.*

*A mi tía Marta que siempre ha estado pendiente de mí.*

*A mis vecinos Marta, Giraudio y Gianni que han sido también mi familia.*

*A Reynaldito que fue mi primer profesor de programación.*

*A mi familia en general que se han preocupado por mí.*

*A mi co-tutor Odriel que me llenaba la tesis de comentario cada vez que se la llevaba.*

*A mis amigos que he hecho durante todos estos años de carrera.*

*A todos los profesores que de una forma u otra han contribuido con mi formación.*

*A la Universidad de las Ciencias Informáticas que ha marcado pauta en mi vida.*

*A todos los que han sido parte de mi formación tanto personal como profesional.*

## *Dedicatoria*

*Mi formación y título de Ingeniero va dedicado a las personas que han estado conmigo toda una vida, que me han apoyado en los momentos difíciles y han transitado junto a mí estos 24 años de vida.*

*Muy en especial a mis abuelas: Elia y Evelia que siempre he sido su razón de ser, y desde chiquito me han tratado de dar todos los gustos. Que siempre han estado orgullosas de mí en todas las decisiones que he tomado. A mi abuela mamá (Elia) que ha sido apoyo de mi mamá en estos años que he estado fuera de casa. A mi abuela Evelia que siempre ha estado al tanto de mí y de mi formación. Mis abuelas, con todo mi corazón les dedico este trabajo y mi formación.*

*A mi madre: Vivian M. Couce Herrera que desde pequeño me ha ayudado, desde que comencé mis estudios siempre ha estado pendiente de mi formación; ha pasado trabajo y sacrificio para que hoy yo esté donde estoy; se ha cohibido de gustos para*

*dármelos a mí; siempre me ha apoyado en todas mis decisiones; ha estado allí cuando la necesito sin pedir nada a cambio; ha sufrido cuando he tenido un problema y ha llorado cuando me he ido a estudiar lejos de ella. A ella, madre incondicional, va dedicado este trabajo.*

*A mi padre: Pedro Fidel Hernández León que desde pequeño ha estado al lado mío, siempre me ha aconsejado y me ha ayudado a salir adelante; siempre ha estado orgulloso de mí y de mis logros; cuando he tenido un problema me ha ayudado a resolverlo; siempre me ha mostrado el camino, ayudándome en mi formación, tanto personal como profesional; un padre, mi amigo, va dedicado este trabajo.*

## *Resumen*

La Universidad de las Ciencias Informáticas (UCI) posee una aplicación que hace una representación geoespacial del entorno universitario denominado Sistema de Información Geográfica de la UCI (SIGUCI) el cual está concebido estructuralmente como un sistema Web; esta aplicación cuenta con desventajas ya que no puede ser usada al no ser a través de una computadora. Para resolver este inconveniente se realizó una personalización de esta aplicación para dispositivos móviles, y poder acceder a ella desde cualquier lugar, en cualquier instante de tiempo, pero la misma no cuenta con todas las funcionalidades del SIGUCI. En este trabajo se perfeccionó la aplicación del SIGUCI para dispositivos móviles agregándole nuevas funcionalidades haciéndola de esta forma más útil y más funcional. Para cumplir esta tarea se estudiaron los Sistemas de Información Geográficos junto a sus diversas aplicaciones, así como los dispositivos móviles con sus tecnologías asociadas, teniendo así un mayor conocimiento del tema en cuestión. En el transcurso de la investigación se detalló la documentación y los artefactos generados según la metodología de *software* escogida, logrando de esta manera la modelación de la aplicación. Como resultado final del trabajo se obtuvo el cumplimiento del objetivo con la elaboración del *software* y la realización de pruebas para chequear su funcionalidad.

**Palabras claves:** Sistema de Información Geográfica, dispositivo móvil, *software*.



## *Ilustraciones*

<i>Ilustración 1 Comparación GeoServer vs MapServer (36)</i>	28
<i>Ilustración 2 Diagrama de clases del Modelo de Dominio</i>	31
<i>Ilustración 3 Diagrama de casos de usos del sistema</i>	36
<i>Ilustración 4 Diagrama de clase del diseño del Caso de Uso Medir Distancia</i>	42
<i>Ilustración 5 Clases persistentes del modelo entidad-relación</i>	43
<i>Ilustración 6 Clases persistentes de la información socio-económica</i>	43
<i>Ilustración 7 Modelo entidad-relación de la base de datos</i>	44
<i>Ilustración 8 Tabla de relación socio-económica</i>	44
<i>Ilustración 9 Vista general del diagrama de componentes</i>	46
<i>Ilustración 10 Vista detallada del Modelo</i>	47
<i>Ilustración 11 Vista detallada de la Vista</i>	47
<i>Ilustración 12 Vista detallada del Controlador</i>	48
<i>Ilustración 13 Modelo de despliegue</i>	49
<i>Ilustración 14 Diagrama de clase del diseño del Caso de Uso Mostrar Información</i>	57

## *Tablas*

<i>Tabla 1 Actor del sistema</i>	35
<i>Tabla 2 Descripción de la tabla tb_socioeconómica de la base de datos</i>	45
<i>Tabla 3 Descripción de la tabla tb_categoría de la base de datos</i>	45
<i>Tabla 4 CU Medir Distancia</i>	54
<i>Tabla 5 CU Mostrar Información</i>	56
<i>Tabla 6 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Medir Distancia</i>	59
<i>Tabla 7 Descripción de Variable del Caso de Uso Medir Distancia</i>	60
<i>Tabla 8 Matriz de Datos Medir Distancia</i>	61
<i>Tabla 9 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Medir Distancia</i>	63
<i>Tabla 10 Descripción de Variable del Caso de Uso Medir Distancia</i>	64
<i>Tabla 11 Matriz de Datos Medir Distancia</i>	65

# Índice

<i>Introducción</i> .....	1
<i>Capítulo I</i> .....	5
<i>Fundamentación Teórica</i> .....	5
1.1    Introducción .....	5
1.2    Conceptos asociados al dominio del problema.....	5
1.3    Objeto de Estudio.....	8
1.3.1    Descripción General .....	8
1.3.2    Descripción actual del dominio del problema .....	9
1.3.3    Situación Problemática .....	9
1.4    Sistema de Información Geográfica para la UCI en dispositivos móviles...	10
1.5    Conclusiones .....	11
<i>Capítulo II</i> .....	12
<i>Tendencias y tecnologías actuales a desarrollar</i> .....	12
2.1    Introducción .....	12
2.2    Las Tecnologías de la Información y las Comunicaciones (TIC) .....	12
2.3    La tecnología móvil .....	13
2.3.1    Protocolo de aplicaciones inalámbricas. Tecnologías WAP .....	13
2.4    Tecnologías y Lenguaje XML .....	14
2.5    Servidor Web .....	15
2.5.1    Servidor Apache .....	16
2.6    Metodologías de desarrollo de <i>software</i> .....	16
2.6.1    Rational Unified Process (RUP).....	17
2.6.2    Extreme Programming (XP).....	18
2.6.3    Metodología de desarrollo escogida .....	18
2.7    Lenguajes de modelado .....	19
2.8    Herramienta de modelado.....	19
2.8.1    Visual Paradigm .....	19
2.8.2    Rational Rose .....	20
2.8.3    Herramienta de modelado escogida .....	21
2.9    Lenguaje de programación.....	21
2.9.1    Lenguaje PHP .....	21
2.10    Entorno de desarrollo integrado (IDE) .....	23
2.10.1    NetBeans.....	23

2.10.2	Eclipse.....	24
2.10.3	Entornos integrados de desarrollo escogido .....	24
2.11	Sistema gestor de base de datos (SGBD).....	24
2.11.1	PostgreSQL.....	25
2.12	Servidor de mapa.....	26
2.12.1	GeoServer .....	26
2.12.2	MapServer .....	26
2.12.3	GeoServer vs MapServer .....	27
2.12.4	Servidor de mapas escogido .....	29
2.13	Conclusiones .....	29
<i>Capítulo III.....</i>		30
<i>Presentación de la solución propuesta. ....</i>		30
3.1	Introducción .....	30
3.2	Entorno donde trabajará el sistema.....	30
3.2.1	Diagrama de clases del Modelo de Dominio .....	30
3.2.2	Glosario de Términos del Dominio.....	31
3.3	Requisitos Funcionales (RF) .....	32
3.4	Requisitos No Funcionales (RNF) .....	34
3.5	Descripción del Sistema Propuesto.....	35
3.5.1	Descripción de los actores.....	35
3.5.2	Diagrama de Casos de Uso del Sistema .....	36
3.5.3	Descripción del Diagrama de Caso de Uso .....	36
3.6	Conclusiones .....	37
<i>Capítulo IV.....</i>		38
<i>Construcción de la solución propuesta. ....</i>		38
4.1	Introducción .....	38
4.2	Patrón de arquitectura de <i>software</i> .....	38
4.2.1	Patrón modelo-vista-controlador (MVC).....	38
4.3	Patrones de diseño .....	39
4.3.1	Patrones generales de <i>software</i> para asignación de responsabilidades. Patrones GRASP .....	39
4.4	Modelo de diseño.....	41
4.4.1	Diagrama de clases del diseño para el Caso de Uso Medir Distancia....	42
4.5	Modelo de datos.....	42
4.5.1	Diagrama de clases persistentes .....	43

4.5.2	Modelo Entidad-Relación.....	44
4.5.3	Descripción de las tablas tb_socioeconómica y tb_categorías de la base de datos	45
4.6	Modelo de implementación .....	46
4.6.1	Diagrama de Componente.....	46
4.7	Modelo de despliegue .....	48
4.8	Diseño de pruebas al sistema .....	49
4.8.1	Pruebas de caja negra.....	49
4.9	Conclusiones .....	50
	<i>Conclusiones generales</i> .....	51
	<i>Recomendaciones</i> .....	52
	<i>Anexos</i> .....	53
	<i>Bibliografías</i> .....	66

## *Introducción*

Los Sistemas de Información Geográfica (SIG) se han convertido durante los últimos años en una de las herramientas más importantes de trabajo para algunos investigadores, analistas y planificadores que desarrollan temas relacionados con geografía. Estos sistemas proveen de funcionalidades para trabajos sobre el terreno geográfico, contribuyendo al análisis y a la toma de decisiones. Los SIG están compuestos por *software*, *hardware* y datos geográficos, estos son una herramienta esencial para el análisis en muchas áreas del desarrollo nacional e internacional para la búsqueda de objetivos físicos en el terreno, analizar el relieve de una zona determinada, conseguir el camino mínimo o más propicio para transitar, entre otras situaciones que requieran de un análisis geoespacial.

Actualmente la informática está ocupando un eslabón importante en todas las esferas de la sociedad, pues constituye una vía fundamental para optimizar la mayor parte de los procesos empresariales, educativos, informativos, culturales y recreativos que se llevan a cabo. El Ministerio de la Informática y las Comunicaciones (MIC) de Cuba, como programa rector de informatización en el país persigue fomentar el desarrollo de la Industria Cubana de las Tecnologías de la Información, debido al auge que ha tomado la computación en todas las esferas de la sociedad, es necesaria su presencia en casi todos sus sectores. Esto se pone de manifiesto en algunos casos como los que siguen: avances tecnológicos que han revolucionado las ciencias médicas en los últimos años, en las comunicaciones con la aparición de la telefonía digital y en el trabajo cotidiano de todas las empresas y demás organizaciones.

La Universidad de las Ciencias Informáticas (UCI) siendo uno de los sectores principales en la informatización del país tiene entre sus proyectos el desarrollo de una aplicación informática denominada Sistema de Información Geográfica de la UCI (SIGUCI). En esta aplicación se hace una representación geoespacial del entorno universitario mostrando todos los puntos de interés con información socio-económica asociada, el mismo le da la posibilidad al usuario de realizar búsquedas mostrando su localización en el mapa. Este *software* es solo accesible desde las computadoras, por lo que los usuarios no pueden consultar el SIG desde cualquier lugar de la universidad, ni en cualquier instante de tiempo. Debido a esto aumenta la importancia de que los estudiantes, profesores y trabajadores de la universidad puedan acceder a

las funcionalidades del SIGUCI, a pesar de no encontrarse utilizando una computadora, satisfaciendo sus necesidades de localización y orientación en el terreno. Existe una personalización del SIGUCI para dispositivos móviles en la universidad que realiza la opción de localización de estudiantes y edificaciones, pero la misma no cuenta con todas las características ni opciones necesarias como el SIGUCI.

El **problema a resolver** queda formulado de la siguiente forma:

¿Cómo calcular la distancia, acceder y visualizar la información socio-económica de la Universidad de las Ciencias Informáticas desde cualquier lugar, en cualquier instante de tiempo a través de los dispositivos móviles?

Siendo el **objetivo** desarrollar para el Sistema de Información Geográfica de la UCI para dispositivos móviles las opciones de cálculo de distancia, acceso y visualización de su información socio-económica referenciada.

Constituyendo el **objeto de estudio** los Sistema de Información Geográfica para dispositivos móviles.

Enmarcando el **campo de acción** en los Sistema de Información Geográfica para dispositivos móviles basado en los servicios Web estandarizados por la *Open Geospatial Consortium*.

Como guía para el desarrollo se plantea la **idea a defender**:

Con el desarrollo de nuevas funcionalidades para el Sistema de Información Geográfica de la UCI para dispositivos móviles se logrará satisfacer la necesidad de calcular la distancia, acceder y visualizar la información socio-económica de la Universidad de las Ciencias Informáticas desde cualquier lugar, en cualquier instante de tiempo.

Para darle solución al objetivo propuesto del problema planteado, se proponen las tareas siguientes:

## **Tareas de la investigación:**

1. Fundamentación de las tendencias actuales, tecnologías y conceptos más importantes relacionados con Sistemas de Información Geográfica.

2. Identificación de los principales servicios Web estandarizados de la *Open Geospatial Consortium*.
3. Identificación de los principales protocolos y tecnologías para el funcionamiento de los dispositivos móviles.
4. Selección de la metodología de desarrollo de *software* y herramientas que podrían utilizarse en el desarrollo de la aplicación.
5. Elaboración de la documentación técnica asociada a la metodología de desarrollo de *software* escogida, haciendo uso de patrones de diseño.
6. Desarrollo de la aplicación informática sobre protocolo WAP haciendo uso de servicios Web estandarizados de la *Open Geospatial Consortium* que garantice a los usuarios de dispositivos móviles calcular la distancia, acceder y visualizar de forma satisfactoria la información socio-económica de la Universidad de las Ciencias Informáticas.
7. Realización de pruebas al *software* mediante el método de Caja Negra, para validar su funcionalidad.

## **Resultados esperados:**

- ✓ Obtención de los artefactos y la documentación asociada al desarrollo del SIG mediante una metodología y herramienta determinada.
- ✓ Obtención del SIG para dispositivos móviles utilizando los servicios Web estandarizados por la *Open Geospatial Consortium*, logrando el cálculo de la distancia y la representación geográfica en un mapa de distintos puntos con informaciones socio-económicas asociada a la Universidad de las Ciencias Informáticas.

A lo largo del desarrollo del *software* se pretende obtener información relevante mediante la aplicación de los siguientes métodos científicos:

## **Métodos teóricos**

- Analítico–sintético: este método se utilizó para analizar y comprender el funcionamiento de algunos de los SIG más utilizados en el mundo y así poder extraer los elementos más importantes para el objeto de estudio.

- Análisis histórico–lógico: este método permitió realizar análisis históricos de algunos de los SIG más utilizados en el mundo, posibilitando el estudio de su evolución para una mejor comprensión de los mismos.
- Modelación: este método permitió un mayor entendimiento del trabajo de diploma mediante la realización del análisis y diseño de la realidad objetiva, destacando sus funcionalidades mediante los principales modelos y diagramas.

Este trabajo de diploma está compuesto por 4 capítulos que abarcan toda la investigación, herramientas, modelación y desarrollo del producto. En el capítulo 1, se desarrolla la “Fundamentación Teórica”, se explican los elementos teóricos que conforman como un todo el problema científico planteado, conjuntamente con los objetivos de trabajo. En el capítulo 2, titulado “Tendencias y Tecnologías Actuales”, se enfatiza acerca de un conjunto de tecnologías y herramientas de formas comparativas que serán necesarias para el desarrollo de la aplicación, se llega a una determinación y se definen cuales se utilizarán. El capítulo 3, denominado “Presentación de la Solución Propuesta”, muestra la descripción del trabajo resultante después de un exhaustivo análisis con las diferentes descripciones y diagramas, mostrando de este modo la modelación de la aplicación. El capítulo 4, que recibe el nombre de “Construcción de la Solución Propuesta”, se aborda todo el proceso de modelación de estos flujos de trabajo, la estructura de la aplicación y demás componentes, así como las pruebas de *software* realizadas a la misma.



## *Capítulo I*

### *Fundamentación Teórica.*

#### **1.1 Introducción**

En la actualidad los Sistemas de Información Geográfica (SIG) y los dispositivos móviles han alcanzado un desarrollo significativo, por lo que ya es una realidad la instalación de los SIG en estos dispositivos, cuyo objetivo principal es facilitar el acceso a información geográfica sin necesidad de consultar enciclopedias, libros y atlas. En este capítulo se abordarán definiciones y conceptos relacionados con los SIG, los servicios Web estandarizados por la *Open Geospatial Consortium*, los dispositivos móviles y las diferentes tecnologías asociadas a estos artefactos. Estos temas son de gran interés para el desarrollo de este trabajo, siendo el punto de partida para comprender la investigación.

#### **1.2 Conceptos asociados al dominio del problema**

Para lograr un mejor entendimiento de los temas que serán abordados en la investigación, se hace necesario relacionar a continuación un conjunto de conceptos asociados a los mismos.

**Sistemas de Información Geográfica:** SIG o GIS, en su acrónimo inglés (*Geographic Information System*) es una integración organizada de *hardware*, *software* y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. Estos sistemas no solo pueden

# CAPÍTULO 1: Fundamentación Teórica

---

ser desplegados en las computadoras; también son utilizados en los dispositivos móviles. (1)

**Dispositivos móviles:** también conocidos como computadora de mano, son de pequeño tamaño, con algunas capacidades de procesamiento, móviles o no, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales. Entre los dispositivos móviles más utilizados están los referentes a la telefonía móvil. (2)

**Teléfono móvil:** es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, cada una de las cuales es una célula. (3)

**Dispositivos Inalámbricos:** es aquel que es capaz de comunicarse o acceder a una red sin cables, por ejemplo, un teléfono móvil, comunicadores de bolsillos o *Personal Digital Assistant* (PDA). Este tipo de dispositivos se comportan como si estuvieran directamente conectados a una red mediante un cable, dando la impresión al usuario de que los datos están almacenados en el propio dispositivo. Esto se logra mediante el uso de los servicios GSM, GPRS y el protocolo WAP. (4)

**Servicio GSM:** son las siglas de *Global System for Mobile communications* (Sistema Global para las comunicaciones Móviles), es el sistema de comunicación móvil digital más utilizado y el estándar de facto para teléfonos móviles en Europa. Definido originalmente como estándar Europeo abierto para que una red digital de teléfono móvil soporte, voz, datos y mensajes de texto en varios países. El GSM es ahora uno de los estándares digitales inalámbricos de la segunda generación más importante del mundo. El GSM está presente en más de 160 países y según la asociación GSM, tienen el 70 por ciento del total del mercado móvil digital. (5)

**Protocolo WAP:** *Wireless Application Protocol* (protocolo de aplicaciones inalámbricas) es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, por ejemplo el acceso a servicios de Internet desde un teléfono móvil. Se trata de la especificación de un entorno de aplicación y de un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos se conectan a Internet. (6)

# CAPÍTULO 1: Fundamentación Teórica

---

---

**Servicio GPRS:** *General Packet Radio Service* (GPRS) o servicio general de transmisión de paquetes vía radio es una extensión del Sistema Global para Comunicaciones Móviles o GSM para la transmisión de datos no conmutada (o por paquetes). Existe un servicio similar para los teléfonos móviles que permite velocidades de transferencia de 56 a 144 kbps. (7)

**Open Geospatial Consortium (OGC):** fue fundado en 1994 para hacer de la información geográfica una parte integral de la infraestructura mundial de información. Es un consorcio internacional, formado por un conjunto de empresas, agencias gubernamentales y universidades, dedicado a desarrollar especificaciones de interfaces para promover y facilitar el uso global de la información espacial. De todos los servicios *web* de OGC, el más conocido y usado es el servicio de mapas o *Web Map Service* (WMS). A pesar de su éxito, muchas de las implementaciones existentes tienen muchos problemas para trabajar con capas de gran resolución o de gran volumen de datos. (8)

Los estándares más importantes surgidos del OGC son:

**GML** - *Geography Markup Language* contiene información específica sobre cómo se debe hacer la visualización de los datos representados, para ello utilizan estilos que se relacionan y se describen en otros sub-lenguajes de XML. (8)

**KML** - *Keyhole Markup Language* es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones. (8)

**WFS** - *Web Feature Service* o Servicio de entidades vectoriales que proporciona la información relativa a la entidad almacenada en una capa vectorial (cobertura) que reúnen las características formuladas en la consulta. (8)

**WMS** - *Web Map Service* o Servicio de mapas en la web que produce mapas en formato imagen a la demanda para ser visualizados por un navegador web o en un cliente simple. (8)

**WCS** - *Web Coverage Service* proporciona una interfaz que permite realizar peticiones de cobertura geográfica a través de la web utilizando llamadas independientes de la plataforma. (8)

**CSW** - *Web Catalogue Service* define una interfaz común para el descubrimiento, búsqueda y consulta de metadatos relacionados a datos, servicios y recursos de tipo geográfico. (8)

## 1.3 Objeto de Estudio

Para el desarrollo de la investigación se definió como objeto de estudio los Sistema de Información Geográfica para dispositivos móviles. Para su confección es preciso profundizar en el estudio de los SIG y de los dispositivos móviles así como sus tecnologías asociadas.

### 1.3.1 Descripción General

Los dispositivos móviles son de uso personal y por lo general su propietario siempre lo trae consigo, por lo que lo adapta de acuerdo a sus intereses y comodidades. Con la aparición de la telefonía móvil digital, fue posible acceder a páginas de Internet especialmente diseñadas para móviles, mediante la tecnología WAP. Las primeras conexiones se efectuaban mediante una llamada telefónica a un número del operador a través de la cual se transmitían los datos de manera similar a como lo haría un módem en una *Personal Computer* (PC).

Posteriormente, nació el GPRS, que permitió acceder a Internet a través del protocolo TCP/IP. Mediante el *software* adecuado es posible acceder, desde una terminal móvil, a servicios como *File Transfer Protocol* (FTP), Telnet, mensajería instantánea, correo electrónico; utilizando los mismos protocolos que un ordenador convencional. La velocidad del GPRS es de 54 kbit/s en condiciones óptimas, y se tarifa en función de la cantidad de información transmitida y recibida. (9)

Los SIG funcionan como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos, también preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

Dentro de los SIG existentes en el mundo muchos pueden ser representados en dispositivos móviles, y por esta razón, estos ofrecen una nueva etapa en su evolución. Los SIG para dispositivos móviles tienen las siguientes utilidades:

- Pueden ser consultados en cualquier lugar y circunstancia.
- Consultan mapas en forma digital.
- Acceden en tiempo real a mapas e informaciones existentes.
- Reemplazan mapas en papel, siendo compacto y pudiendo almacenar gran cantidad de información.
- Ayudan en la tomar de decisiones desde el terreno.

### 1.3.2 Descripción actual del dominio del problema

La Universidad de la Ciencias Informáticas ha alcanzado un tamaño significativo, por tal motivo, se realizó un mapa del entorno en formato de imagen para orientar y ubicar a los nuevos ingresos. Con el desarrollo de los proyectos y el trabajo con los SIG se realizó el Sistema de Información Geográfica de la UCI (SIGUCI). Este *software* dio cobertura a que los usuarios pudieran visualizar y buscar desde la web en el entorno universitario. Entre sus funcionalidades están las opciones básicas de un SIG (zoom, paneo, centrar mapa y selección de capas), opciones de búsquedas temáticas, mostrar una información descriptiva de un punto en el mapa, así como visualizar su localización, medición de distancia para que el usuario conozca una estimación del recorrido, entre otras opciones. Luego de un análisis del SIGUCI se comienzan a identificar desventajas en su utilización, entre ellas la necesidad de una estación de trabajo fija para su uso.

### 1.3.3 Situación Problemática

Debido a que durante los últimos años los dispositivos móviles han progresado en cuanto a su uso por la sociedad y en cuanto a su desarrollo tecnológico incorporado a su sistema *software* de todo tipo, muchas empresas se han dedicado a la elaboración de programas informáticos para estos dispositivos. Con el avance de los SIG y su gran utilización en las distintas esferas se han comenzado a realizar personalizaciones para los móviles y de esta forma solucionar problemas en el propio terreno, apresurando el trabajo y la búsqueda de información. Existe una diferencia entre los SIG convencionales y los SIG para dispositivos móviles, estos últimos tienen la ventaja de poder ser consultados por el usuario en cualquier momento, independientemente del lugar donde se encuentre el mismo, a diferencia de los primeros que para ser consultados es necesario trasladarse hacia el lugar donde se encuentre la computadora que contiene el *software* instalado.

# CAPÍTULO 1: Fundamentación Teórica

---

---

La Universidad de las Ciencias Informáticas (UCI) tiene un tamaño significativo y todos los años entran nuevos estudiantes y trabajadores al centro que no conocen todos los lugares de la universidad. La UCI cuenta con un sin número de cafeterías, plazas, salidas de pases, docentes, edificaciones, postas médicas, entre otras estructuras que necesitan ser localizadas y conocer la distancia entre ellas, así como su información socio-económica, por cualquier individuo interno o trabajador externo de la universidad. Ninguna de estas personas cuenta con mecanismos de orientación en el terreno universitario tanto para localizar estos sitios como para buscar los datos de las personas que habitan la residencia.

La UCI tiene una aplicación informática denominada Sistema de Información Geográfica de la UCI (SIGUCI) que puede resolver este inconveniente, pero esta aplicación es solo accesible desde una computadora. Dicha aplicación no es accesible desde cualquier lugar de la universidad, ni en cualquier instante de tiempo. Por tal motivo no resuelve las necesidades de localización y orientación en el terreno.

## **1.4 Sistema de Información Geográfica para la UCI en dispositivos móviles**

El desarrollo de los SIG para móviles en Cuba es una nueva línea de investigación, así como la programación en dispositivos móviles. En la Universidad de las Ciencias Informáticas se desarrolló un sistema que a pesar de no estar en funcionamiento hace una representación del SIGUCI en dispositivos móviles utilizando los estándares de la OGC. Para su confección se utilizó el lenguaje de programación PHP y MapServer como servidor de mapas. Esta aplicación tiene entre sus funcionalidades las opciones de:

- Aumentar y disminuir la escala del mapa para alejarlo y acercarlo (zoom+ y zoom-).
- Desplazar el mapa hacia arriba, abajo, izquierda y derecha sin variarle la escala (paneo).
- Seleccionar las capas que desee visualizar en el mapa.
- Localizar donde reside una persona en el mapa.
- Buscar una edificación según el área temática a la que pertenezca.
- Realizar la búsqueda de un edificio y representarlo en el mapa.

# CAPÍTULO 1: Fundamentación Teórica

---

---

A pesar de tener varias funcionalidades, este sistema no cuenta con todas las funcionalidades del SIGUCI. Por tal motivo se pretende desarrollar sobre esta aplicación nuevas funcionalidades como el cálculo de distancia y la visualización de información socio-económica asociada para hacerlo más útil y funcional.

## **1.5 Conclusiones**

En este capítulo se analizaron los conceptos de SIG, OGC, Wap y dispositivos móviles que ayudarán a entender los términos técnicos de este trabajo. Se realizó una descripción del objeto de estudio conociendo así el dominio del problema. Con el análisis de las soluciones existentes se llegó a la conclusión que existe una necesidad de realizar un perfeccionamiento del SIG para móviles de la universidad, haciéndolo más útil y funcional. Una vez concluido este capítulo se hará un análisis de las tecnologías y herramientas que intervendrán en la realización de la aplicación.

### *Capítulo II*

## *Tendencias y tecnologías actuales a desarrollar.*

### **2.1 Introducción**

Actualmente el desarrollo de las tecnologías de la informática es cada vez más elevado; esto ha traído consigo que exista una gran cantidad de herramientas de *software* para minimizar el trabajo de desarrollo de una aplicación informática, lo que ha proporcionado competencias entre los principales proveedores de estas herramientas. Por tal motivo para poder desarrollar de forma eficiente un producto informático es necesario hacer un análisis de las ventajas, desventajas y cualidades de una determinada herramienta de *software*. También se identifican las tecnologías y metodologías a utilizar para su desarrollo.

### **2.2 Las Tecnologías de la Información y las Comunicaciones (TIC)**

En parte, estas nuevas tecnologías son inmateriales, ya que la materia principal es la información; permiten la interconexión y la interactividad; son instantáneas; tienen elevados parámetros de imagen y sonido. Al mismo tiempo las nuevas tecnologías suponen la aparición de nuevos códigos y lenguajes, la especialización progresiva de los contenidos sobre la base de la cuota de pantalla (rompiendo la cultura de masas) y dando lugar a la realización de actividades inimaginables en poco tiempo. (10)

Sus principales aportaciones a las actividades humanas se concretan en una serie de funciones que facilitan la realización de nuestros trabajos, para cualquier caso, siempre requieren una cierta información para realizarlo, un determinado proceso de datos y a menudo también la comunicación con otras personas. Entre los principales aportes que ofrecen las TIC están:

- Fácil acceso a todo tipo de información.
- Instrumentos para todo tipo de proceso de datos.
- Canales de comunicación.
- Almacenamiento de grandes cantidades de información.
- Automatización de tareas.



## CAPÍTULO 2: Tendencias y tecnologías actuales

---

- Interactividad.
- Homogeneización de los códigos. (11)

### 2.3 La tecnología móvil

La evolución de los dispositivos móviles, aparejada a la de la telefonía móvil, en los últimos años ha sido muy significativa. Hoy en día cualquier dispositivo móvil cuenta con acceso a la red telefónica inalámbrica celular y se han creado para estos dispositivos nuevas tecnologías que permiten una mejor comunicación a Internet. (12)

Las redes actuales de telefonía móvil permiten velocidades medias competitivas en relación con las de banda ancha en redes fijas: 183 kbps en las redes GSM, 1064 kbps en la tercera generación y 2015 kbps en las *WiFi*. (13)

Las primeras tecnologías que permitieron el acceso a datos aunque a velocidades no excesivas fueron el GPRS y el EDGE (*Enhanced Data rates for GSM of Evolution*), ambas pertenecientes a lo que se denomina segunda generación. Aunque, la banda ancha en telefonía móvil empezó con la tercera generación que permitía 384 kbps, HSPA (*High Speed Packet Access*) que permite hasta 14 Mbps de bajada HSDPA (*High Speed Downlink Packet Access*) y teóricamente 5,76 Mbps de subida si se utiliza a más HSUPA (*High Speed Uplink Packet Access*); estas velocidades son, en ocasiones, comparables con las xDSL, y en un futuro no muy lejano también se prevé que empiecen a estar disponibles tecnologías más avanzadas, denominadas genéricamente *Long Term Evolution* o redes de cuarta generación y que permitirán velocidades de 50 Mbps. (14)

#### 2.3.1 Protocolo de aplicaciones inalámbricas. Tecnologías WAP

La tecnología WAP es realmente un estándar impulsado por la industria del sector de las telecomunicaciones con el objetivo de proporcionar un sistema avanzado de servicios de internet para dispositivos móviles. La tecnología tiene como premisas iniciales el uso de estándares abiertos ya existentes (como los protocolos HTTP, o el XML), la independencia de la tecnología de comunicaciones móviles sobre la que se implemente (en principio, GSM, pero en el futuro podría ser GPRS o incluso UMTS) y la independencia del terminal móvil (desde un teléfono hasta un PDA). (15)

A partir de las premisas anteriores, el conjunto de protocolos incluidos en el estándar WAP debe adaptarse a las condiciones propias de un entorno totalmente nuevo, como

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

es el de las comunicaciones móviles. En concreto, la red de comunicaciones presenta limitaciones de ancho de banda importantes, alta latencia y cobertura intermitente, y los terminales de acceso no tienen grandes capacidades de procesamiento, ni de memoria, además de utilizar pantallas que no permiten más de 4 o 5 líneas de texto. (15)

La tecnología WAP se basa en 3 elementos:

- Cliente: el cliente será el micro-navegador WML de un terminal móvil. Este será el equivalente al navegador de un PC en el internet de computadora.
- Pasarela: la pasarela constituye la interfaz entre la red inalámbrica y la red física. Es lo que se conoce como el *WAP Gateway*.
- Servidor Web: es un servidor encargado de procesar las peticiones del cliente y enviar las páginas solicitadas. (15)

El cliente o dispositivo WAP está provisto de un micro-navegador que constituye la interfaz de usuario para realizar las funciones de navegación. El micro-navegador interpreta páginas WML. El WML sería el equivalente al HTML del internet que todos conocen. (15)

La pasarela realiza 2 funciones básicas: conversión de protocolos (de HTTP a WSP/WTP y viceversa) y codificación / decodificación de las páginas WML. Estos procesos permiten la adaptación a la red inalámbrica del protocolo y de los contenidos. (15)

Finalmente, en el servidor web recibe las páginas, así como cualquier otra lógica basada en CGI, acceso a bases de datos o lenguajes de script. WAP es compatible con servidores HTTP 1.1, lo que facilita la adopción del estándar por parte de los proveedores de contenidos web ya existentes. (15)

### **2.4 Tecnologías y Lenguaje XML**

XML (lenguaje de marcas extensible) es una tecnología que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Las tecnologías XML son un conjunto de módulos que ofrecen servicios a

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. (16)

XML es también un lenguaje de etiquetado del lado del cliente muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. (16)

Ventajas de XML:

- Es extensible: después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Se puede comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, se podría tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- Transforma datos en información, pues se le añade un significado concreto y se asocia a un contexto, con lo cual existe flexibilidad para estructurar documentos. (16)

### 2.5 Servidor Web

Un servidor Web es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir lo que se llama en hipertextos, páginas Web o página HTML textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. (17)

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se suele conocer como navegador. El navegador realiza una petición al servidor y este le responde con el contenido que el cliente solicita. Un servidor web es un programa informático que procesa una aplicación del

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. (17)

### 2.5.1 Servidor Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido. (18)

#### Características del Servidor Apache

- Los módulos Apache API se utilizan en un nuevo conjunto de interfaces de programación de aplicaciones (APIs).
- Los módulos pueden actuar como filtros de contenido.
- Soporta la próxima generación de formato de direcciones IP.
- Se han eliminado una serie de directrices complicadas y otras se han simplificado.
- Cuando usa documentos *Server Side Include* (SSI), las páginas de errores personalizables se pueden entregar en diversos idiomas. (18)

La Fundación de *Software* Libre no considera a la Licencia Apache como compatible con la versión 2 de la GNU *General Public License* (GPL). Sin embargo, la versión 3 de la GPL incluye una provisión (Sección 7e) que le permite ser compatible con licencias que tienen cláusulas de represalia de patentes, incluyendo a la Licencia Apache. (19)

Apache es usado por muchas aplicaciones gracias a los servicios que puede brindar y su capacidad de funcionar en diversos sistemas operativos, fue desarrollado con la tendencia de *software* libre, la cual es de vital importancia para Cuba. Para el desarrollo de la aplicación se seleccionó el servidor web Apache 2.x, ya que es un servidor que soporta sin problemas la aplicación y tiene compatibilidad para Wap.

### 2.6 Metodologías de desarrollo de *software*

Las metodologías de desarrollo de *software* son un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en un *software*. Cada metodología de desarrollo de *software* tiene su propio enfoque para su ciclo de vida. A

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

la hora de seleccionar una metodología la primera pregunta que se plantea es: ¿Una metodología ágil o una metodología pesada? La gran mayoría de proyectos se pueden beneficiar del uso de una metodología ágil dado que requiere de menor tiempo para su documentación, pero indudablemente existen proyectos y entornos en los que es una condición, generalmente impuesta por el cliente o la dirección de la empresa, que el proyecto se desarrolle con más documentación. Actualmente existen muchas metodologías que están caracterizadas en distintas tendencias de desarrollo de *software*, entre ellas se tiene la de Programación Extrema siendo entre las más populares de las metodologías ágiles y de las metodologías pesadas la más popular es el Proceso Unificado Racional, a continuación se exponen sus principales características.

### 2.6.1 Rational Unified Process (RUP)

*Rational Unified Process* (RUP) o Proceso Unificado Racional es una metodología pesada de desarrollo de *software*. Tiene una infraestructura flexible de desarrollo de *software* que proporciona prácticas recomendadas probadas y una arquitectura configurable, es un proceso práctico. RUP es una metodología de desarrollo de *software* para la Ingeniería del *Software* Orientada a Objetos, proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo de *software*. Define, para cada etapa y cada disciplina, el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen. Su meta es asegurar la producción de *software* con la más alta calidad, que reúna las necesidades de los usuarios dentro del cronograma planeado y la inversión prevista. RUP deja bien definido y estructurado el proceso de diseño de un *software*. Define claramente quien es responsable, cómo se hacen las cosas y cuando hacerlas. (20)

Mantiene una estructura bien definida del ciclo de vida de un proyecto y, de forma general, está caracterizado por los siguientes aspectos:

- Dirigido por los casos de uso: los casos de uso son el instrumento para validar la arquitectura del *software* y extraer los casos de prueba.
- Centrado en la arquitectura: los modelos son proyecciones del análisis y el diseño, constituye la arquitectura del producto que se desea desarrollar.

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

---

- Iterativo e incremental: durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo. (20)

A través del marco de las mejores prácticas, el Proceso Unificado Racional promueve una cultura y una visión común al interior de su organización de desarrollo, lo que facilita la comunicación entre miembros del equipo, factor clave en el éxito de un proyecto. El proceso ayuda en la definición y administración de las actividades de desarrollo y provee, a los líderes de proyectos, guías específicas para asegurar una administración exitosa del desarrollo iterativo. (20)

### 2.6.2 Extreme Programming (XP)

*Extreme Programming* (XP) o Programación Extrema es una metodología ágil y es la que ha recibido más atención de todas ellas. Construye un proceso de diseño evolutivo. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. El resultado es un proceso de diseño disciplinado, lo que es más, combina la disciplina con la adaptabilidad de una manera que indiscutiblemente la hace la más desarrollada entre todas las metodologías adaptables. Entre las cosas más llamativas es su fuerte énfasis en las pruebas. XP establece la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en el proceso de integración continua y construcción lo que rinde una plataforma altamente estable para el desarrollo futuro. Los valores originales de la programación extrema son: simplicidad, comunicación, retroalimentación (feedback) y coraje. Un quinto valor, respeto, fue añadido en la segunda edición de *Extreme Programming Explained*. (21)

### 2.6.3 Metodología de desarrollo escogida

Para el desarrollo de la aplicación la metodología escogida es RUP, ya que define para cada etapa y cada disciplina, el flujo de trabajo, los trabajadores que intervienen, las actividades que realizan y los artefactos que se necesitan o producen; posibilitando de esta manera que quede una documentación bien detallada. Esto trae como beneficio la posibilidad de dar seguimiento en un futuro para actualizar la aplicación y además ofrece un seguimiento al desarrollo del *software*.

### 2.7 Lenguajes de modelado

*Unified Modeling Language* (UML) o Lenguaje Unificado de Modelado es un lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (22)

Es importante resaltar que UML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. Se puede aplicar en el desarrollo de *software* entregando gran variedad de formas para dar soporte a una metodología de desarrollo de *software* (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar. (22)

### 2.8 Herramienta de modelado

Es una aplicación informática que permite aumentar la productividad en el desarrollo de *software*, reduciendo su costo en términos de tiempo y recursos. Estas herramientas pueden facilitar el trabajo en todas las fases del ciclo de vida del *software*. Las herramientas de *Computer Aided Software Engineering* (CASE) o de modelado son un conjunto de métodos, utilidades y técnicas que dan asistencia a los analistas y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un *software*, facilitando el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

#### 2.8.1 Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Dentro de sus principales características se tienen:

- Entorno de creación de diagramas para UML 2.x.

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

- Diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDE.
- Disponibilidad en múltiples plataformas. (23)

Este *software* de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (23)

### 2.8.2 Rational Rose

Es actualmente conocida como una familia de *software* de IBM para el despliegue, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de *software*. Proporciona herramientas que expanden las prácticas modernas de ingeniería de *software*, particularmente la arquitectura modular y el desarrollo iterativo. Rational Rose aporta un lenguaje común de modelado para el equipo que facilita la creación de *software* de calidad más rápidamente. (24)

Características adicionales incluidas:

- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería *Forward* y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo - código configurables.
- Soporte Enterprise Java Beans 2.0.
- Capacidad de análisis de calidad de código.
- El *Add-In* para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.



## CAPÍTULO 2: Tendencias y tecnologías actuales

---

- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos. (24)

Esta es una herramienta muy potente para la ingeniería de *software*, con ella se podría desarrollar satisfactoriamente el modelado de la aplicación, pero tiene como desventajas que es *software* propietario.

### 2.8.3 Herramienta de modelado escogida

Como herramienta de modelado se escoge Visual Paradigm for UML 8.0 Enterprise Edition, por ser una potente herramienta, tener soporte para el desarrollo de RUP (metodología de trabajo escogida). Es una aplicación multiplataforma, a diferencia del Rational Rose, y facilita el desarrollo de todo el ciclo de vida del *software*. A pesar de no ser *software* libre tiene su licencia avalada por la universidad.

## 2.9 Lenguaje de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación. También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos. (25)

### 2.9.1 Lenguaje PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. (26)

Ventajas:

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5). (26)

Con este lenguaje se puede programar sin dificultad para dispositivos móviles y existe librerías XML llamadas WURFL ( *Wireless Universal Resource File*) y WALL ( *Wireless Abstraction Library*) abaladas por un proyecto de código abierto donde se detallan las características y capacidades de la gran mayoría de dispositivos móviles existentes en el mercado. En esta librería se proporcionan características generales del dispositivo como el modelo, la marca, características de la pantalla (ancho, alto, filas, columna), capacidad de procesamiento de imágenes (soporte para gif, flash lite, colores de pantalla). Se puede saber cosas tan interesantes como el tamaño máximo de la url ( *Uniform Resource Locator*) que puede procesar.

Para la implementación del *software* se escoge PHP 5 debido a que este trabajo de diploma tributa directamente al desarrollo de la línea de SIG para dispositivos móviles siguiendo como base el proyecto de investigación y desarrollo de SIGUCI, su código fuente se encuentra programado en PHP, y por ende no existiría incompatibilidad de lenguaje. PHP 5 es un lenguaje muy utilizado en la tecnología WAP, tiene bastante documentación y una gran comunidad que lo respalda, no necesita ninguna tecnología

con gran capacidad de memoria para su desarrollo y se pueden programar de forma eficaz todas las funcionalidades necesitadas para la aplicación.

### 2.10 Entorno de desarrollo integrado (IDE)

Es un programa informático compuesto por un conjunto de herramientas de programación. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Algunos IDE soportan múltiples lenguajes, tales como Eclipse o NetBeans. El soporte para lenguajes alternativos es a menudo proporcionado por *plugins*, que les permite ser instalado en el IDE. Por ejemplo, Eclipse y NetBeans tiene *plugins* para C / C ++, Ada, Perl, Python, Ruby y PHP, entre otros lenguajes.

#### 2.10.1 NetBeans

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento y con socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos. (27)

NetBeans IDE es un entorno de desarrollo integrado, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java y permite la programación en varios lenguajes como PHP. Existe además un número importante de módulos para extender el NetBeans IDE, es un producto libre y gratuito sin restricciones de uso. (27)

Características destacadas:

- Sintaxis resaltada, completamiento de código y análisis de *tipeo*.
- Verificación de Sintaxis.
- Asistentes para configuración de archivos XML y controladores *Spring Web MVC*.
- Menor consumo de memoria.
- Soporte de entorno *Spring Web MVC* en proyectos web.
- Aplicaciones para móviles con documentación y estabilidad. (28)

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

Esta plataforma de desarrollo cuenta con módulos que se le pueden instalar para ampliar su campo de trabajo y agregarle nuevas funcionalidades. Entre las funcionalidades que puede brindar está la programación en el lenguaje PHP y la creación de aplicaciones para dispositivos móviles. Posee excelentes mecanismos de conexión para las bases de datos MySQL y PostgreSQL, así como para la utilización de sus respectivas bases de datos espaciales.

### 2.10.2 Eclipse

Es un IDE de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente liviano" basadas en navegadores. Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para Visual Age. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. (29)

El IDE de Eclipse emplea módulos o *plug-in* para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de *software*. Está diseñado para la programación en Java y le permite extenderse usando otros lenguajes de programación como PHP. (29)

### 2.10.3 Entornos integrados de desarrollo escogido

Para el desarrollo satisfactorio de la aplicación se propone la implementación con el IDE NetBeans 6.9, siendo una plataforma de desarrollo donde se puede programar sin dificultad en el lenguaje PHP, es una potente herramienta y es un producto libre sin restricciones de uso. Cuenta con un módulo para el desarrollo de aplicaciones móviles y tiene documentación para la misma. Cuenta además con una gran comunidad que ofrece soporte a la plataforma. Es una aplicación multiplataforma con versiones para Windows y Linux.

## 2.11 Sistema gestor de base de datos (SGBD)

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (*DataBase Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

### 2.11.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos, es *software* libre publicado bajo la licencia BSD (*Berkeley Software Distribution*). Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de formas desinteresadas, altruistas, libres y apoyadas por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*). (30)

PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. (30)

PostgreSQL es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad, vistas, entre otras opciones. (31)

PostgreSQL cuenta con un modulo llamado PostGIS que le añade soporte de objetos geográficos a la base de datos objeto-relacional, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la Licencia pública general de GNU en su versión 2. (32)

PostGIS ha sido desarrollado por la empresa canadiense Refraction Research, especializada en productos "*Open Source*" entre los que habría que citar a Udig. PostGIS es hoy en día un producto veterano que ha demostrado versión a versión su eficiencia. En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL, y a juicio de muchos, es muy similar a la versión geográfica de la archiconocida Oracle. (32)

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

Se escoge como sistema gestor de base de datos a utilizar PostgreSQL 8.4 porque es un sistema gestor de base de datos libre y no existe ninguna restricción en cuanto a su uso. PostgreSQL 8.4 tiene un módulo (la librería PostGIS) que se le integra para poder almacenar datos espaciales, el cual ha sido comprobado en cuanto a su eficiencia y superioridad respecto a sus homólogas en otros SGBD como MySQL. Posee conexión para el trabajo con PHP y NetBeans. Además es la base de datos utilizada del proyecto de investigación y desarrollo de SIGUCI, donde sus datos se encuentran almacenados en una base de datos existente en PostgreSQL con el modulo PostGIS, y por ende no existiría incompatibilidad.

### 2.12 Servidor de mapa

Los servidores de mapas proveen cartografía a través de la red tanto en modo vectorial como con imágenes. Permiten al usuario la máxima interacción con la información geográfica. El usuario accede a información en su formato original, de manera que es posible realizar consultas tan complejas como las que haría un SIG.

#### 2.12.1 GeoServer

Es un servidor de *software* escrito en Java que permite a los usuarios compartir y editar información geo-espacial. Diseñado para una máxima interoperabilidad, publica datos utilizando estándares abiertos desde las mayores bases de datos espaciales y diversos formatos de archivos tanto vectoriales como *raster*. Al ser un proyecto gestionado por la comunidad, GeoServer es desarrollado, probado y apoyado por un diverso grupo de individuos y organizaciones de todo el mundo. GeoServer es la implementación de referencia de los estándares del *Open Geospatial Consortium* (OGC), *Web Feature Service* (WFS) y *Web Coverage Service* (WCS), así como un *Web Map Service* (WMS) certificado y altamente optimizado. GeoServer constituye un componente esencial en la Web Geoespacial. (33)

#### 2.12.2 MapServer

Es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG en Internet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología *Internet Map Server* (IMS).

Sus características principales son:

## CAPÍTULO 2: Tendencias y tecnologías actuales

---

- Se ejecuta bajo plataformas Linux/Apache y Windows (MS4W).
- Formatos vectoriales soportados: ESRI shapefiles, PostGIS, ESRI ArcSDE, GML y otros muchos vía OGR.
- Formatos *raster* soportados: JPG, PNG, GIF, TIFF/GeoTIFF, EPPL7 y otros vía GDAL.
- Fuentes TrueType.
- Configuración "al vuelo" vía parámetros GET pasados por URL.
- MapScript proporciona una API para poder acceder a las funcionalidades de MapServer mediante lenguajes de programación como PHP, Java, Perl, Python, Ruby o C#. (34)

### 2.12.3 GeoServer vs MapServer

A continuación se realizará una comparación entre estos dos servidores de mapas y en qué caso conviene elegir uno u otro.

**Prestaciones:** MapServer es mucho superior en este aspecto en cuanto a consumo de CPU y memoria RAM. (35)

**Portabilidad:** GeoServer es más portable, se puede correr en Windows, en Mac o en Linux sin mayores modificaciones y si existen versiones compiladas. (35)

**Mantenibilidad:** GeoServer tiene una gran ventaja, debido a que tiene una interfaz de usuario para manejar la configuración y MapServer solo se puede hacer mediante un archivo .map, un solo error en el archivo .map y ya no cargará. (35)

**Cantidad de instancias:** GeoServer no puede filtrar los mapas compartidos por WMS, por lo tanto hay que tener una instancia por cada filtro, aquí MapServer es mejor ya que en lugar de tener X instancias que consumen RAM (como en GeoServer) solo hay que cambiarle en la URL el archivo .map y se tiene una nueva instancia. (35)

**WFS:** ambos soportan WFS pero GeoServer soporta WFS-t (WFS transaccional), lo que significa que se puede modificar el mapa a través de un *webservice*. (35)

#### ¿Cuándo usar cual?

Se debe utilizar MapServer siempre que los recursos (CPU, RAM) sean limitados, o cuando es necesario tener varias instancias de un mismo mapa para compartir los datos por WMS. (35)

## CAPÍTULO 2: Tendencias y tecnologías actuales

En cambio se debe utilizar GeoServer cuando no existen restricciones de recursos, si está en constante modificación muchas cosas de la aplicación (mantenibilidad) o si es prioritario el soporte para wfs-t. (35)

La siguiente imagen muestra algunos criterios de comparación entre estos dos servidores de mapas: (Ver Ilustración 1)

	<b>Geoserver</b>	vs.	<b>Mapserver</b>
<b>WMS</b>	Bueno		Excelente
<b>WFS</b>	Soporta WFS-T		No soporta WFS-T
<b>Tecnología</b>	J2EE		CGI
<b>Comienzo</b>	2003		1996
<b>Administración</b>	Interfaz web		Generación asistida por plugins o desarrollo puro
<b>Extensibilidad</b>	Desarrolladores Java		Desarrolladores PHP
<b>Simbología</b>	SLD's estándar		Estilos parte del MapFile
<b>Servicios</b>	WMS/WFS/WCS		Un mapfile por cada servicio web
<b>Consultas</b>	OGC Filter Encoding y CQL		Sentencias SQL

**Ilustración 1 Comparación GeoServer vs MapServer (36)**

Ahora se presentará un resumen de los principales aspectos que diferencian a estos dos servidores de mapas:

- MapServer generalmente es mejor en el manejo de servicios WMS mientras GeoServer administra mejor servicios WFS
- Una gran diferencia entre ellos es que GeoServer soporta servicios WFS Transaccionales y MapServer no lo hace.
- MapServer trabaja con CGI, GeoServer lo hace con J2EE. Esto puede significar alguna ventaja para GeoServer, ya que algunas compañías no desean trabajar con CGI.
- MapServer es un proyecto más maduro (1996), mientras GeoServer es un poco más reciente (2003)



## CAPÍTULO 2: Tendencias y tecnologías actuales

---

- GeoServer tiene una interfaz web de administración que facilita su configuración.
- MapServer tiene sin lugar a dudas un poderoso sistema cartográfico, que provee datos bajo vectores dinámicos con alta calidad. (36)

### 2.12.4 Servidor de mapas escogido

Como servidor de mapas se propone utilizar MapServer 5.6.6 al ser un sistema de *software* libre y un potente gestor de mapas. También debido a que esta aplicación tiene como base el proyecto de investigación y desarrollo de SIGUCI y la misma usa como servidor de mapas MapServer. Este servidor soporta sin ninguna dificultad los servicios Web estandarizados por la *Open Geospatial Consortium* y tiene una excelente compatibilidad con el WMS siendo uno de los servicios Web de la OGC a utilizar por la aplicación. Es más eficiente en el consumo del CPU y memoria RAM y es un proyecto que lleva varios años de usabilidad. El sistema cartográfico de MapServer es muy poderoso y es recomendable usar MapServer para situaciones donde los recursos sean limitados ya que no sobrecarga la computadora con muchos recursos.

### 2.13 Conclusiones

En este capítulo se realizó un análisis de las principales tendencias y tecnologías actuales, concluyendo que WAP y GPRS serán las más utilizadas en el desarrollo de la aplicación por la parte de conectividad. Se definió la metodología RUP y lenguaje de programación Java para utilizar en el desarrollo del *software*. Después de un análisis se definió las herramientas a utilizar en la aplicación, tratando siempre que fuesen orientadas hacia el *software* libre como el caso del NetBeans 6.9 y en el caso del Visual Paradigm for UML 8.0 Enterprise Edition que tuviera la licencia avalada. Como SGBD se definió que fuese un *software* potente, libre y que tuviera la posibilidad de almacenar datos espaciales, por tal motivo se escogió el PostgreSQL 8.4 con su extensión PostGIS. Por último se hizo una comparativa a fondo del servidor de mapas para la cartografía, se concluyó que el MapServer 5.6.6 satisfacía las demandas necesarias para el objetivo del trabajo. Una vez concluido este capítulo y definidas todas las herramientas y tecnologías es posible comenzar a desarrollar la propuesta del sistema para dicha aplicación informática.

### *Capítulo III*

### *Presentación de la solución propuesta.*

#### **3.1 Introducción**

Para el desarrollo exitoso de la aplicación es necesario identificar los principales requisitos funcionales y no funcionales que debe tener el sistema para posteriormente detallar sus principales funcionalidades. Aspectos como los antes mencionados se comprenden mejor modelándolos mediante el conjunto de normativas y artefactos que proporciona la metodología RUP; por todo esto en el presente capítulo se realizará el modelo de dominio y se modelará el sistema con sus elementos correspondientes.

#### **3.2 Entorno donde trabajará el sistema**

Lo que se denomina como Modelo de Dominio es empleado fundamentalmente cuando los flujos de información son difusos, es decir, que tengan múltiples orígenes y cuando son solo eventos o sucesos. También la imposibilidad de realizar subsistemas en el proceso de desarrollo del *software* dado por las múltiples interconexiones, es uno de los factores que influyen en la decisión de realizar un modelo de este tipo. Además de la existencia del solapamiento o de múltiples responsabilidades y la dificultad en el establecimiento de las reglas del funcionamiento del producto a implementar.

##### **3.2.1 Diagrama de clases del Modelo de Dominio**

Un Modelo del Dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, además de que no incluyen las responsabilidades de las personas que ejecutan las actividades. Según RUP este tipo de modelo representa un subconjunto del Modelo de Objeto del Negocio, lo cual no significa que siempre que haya un Modelo de Dominio tenga que existir obligatoriamente un Modelo de Negocio. Por lo que el Modelo de Dominio no es más que la representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés.

## CAPÍTULO 3: Presentación de la solución propuesta

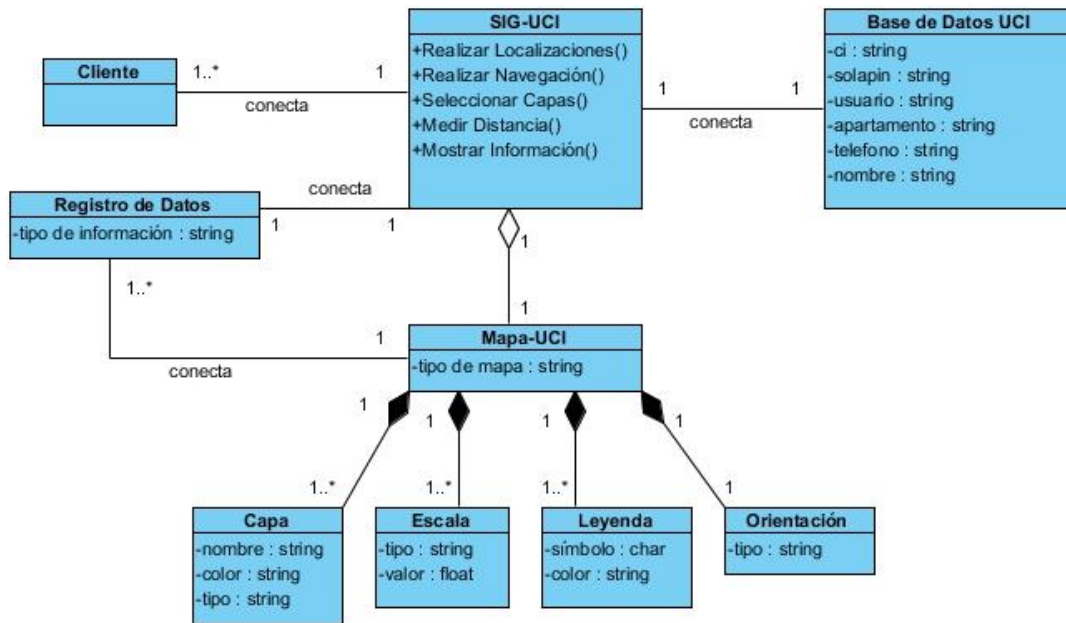


Ilustración 2 Diagrama de clases del Modelo de Dominio

### 3.2.2 Glosario de Términos del Dominio

**Cliente:** personas que se conectan a través de un dispositivo móvil y cuentan con acceso a la red de la Universidad de las Ciencias Informáticas.

**SIG-UCI:** es la entidad donde estarán ubicados los métodos de la aplicación y hay que conectarse para obtener la información. Aquí se realizan todas las funcionalidades del sistema a través de la obtención del flujo de datos.

**Registro de Datos:** es conjunto de datos organizados referentes al aspecto social y económico de cualquier lugar de interés de la universidad e información acerca de la localización de un objeto espacial en un sistema de coordenadas.

**Base de datos UCI:** es el conjunto de datos organizados de todos los estudiantes, profesores y trabajadores de la universidad.

**Mapa-UCI:** se refiere propiamente a la cartografía que mostrará la aplicación una vez que sea utilizada por el usuario. En este caso la representación geográfica de la Universidad de las Ciencias Informáticas.

**Capas:** es un mapa monotemático que se integra a una serie de mapas para formar uno de carácter más específico en un sistema de información geográfica.

## CAPÍTULO 3: Presentación de la solución propuesta

---

---

**Escala:** la escala expresa la razón de ampliación o reducción entre una distancia en el mapa y una distancia equivalente en el terreno. El conocer la escala de un mapa permite medir distancias, determinar áreas y realizar comparaciones entre diferentes objetos.

**Leyenda:** es una guía que le brinda la aplicación al usuario para que le sea fácil trabajar e interpretar los símbolos, los colores, las tramas y los sombreados empleados del mapa.

**Orientación:** es la relación existente entre las direcciones norte, sur, este y oeste ya sea desde un punto de referencia en función de otro o bien la orientación de un elemento geográfico.

### **Breve descripción del diagrama**

El cliente se conecta a través de una terminal móvil al SIG-UCI, quien se conecta a su vez al Registro de Datos que contiene los datos socio-económicos de la información referenciada para cada punto de interés que se encuentra ubicado en el mapa. SIG-UCI se conecta además a la Base de Datos UCI, siendo esta la base de datos de la universidad donde se encuentran ubicados los datos de todos los estudiantes, profesores y trabajadoras. SIG-UCI tiene una relación de agregación con Mapa-UCI que tiene la función de mostrar la cartografía.

El Mapa-UCI está compuesto por varias escalas representativas, leyendas para su comprensión, capas para mostrar los puntos de intereses y la orientación para poder localizar los objetos de su interior; todos estos elementos permiten un mejor entendimiento del mapa. También el Mapa-UCI consulta la información del Registro de Datos.

Los mapas a utilizar los creará el centro GEYSED, siendo la Universidad de las Ciencias Informáticas la que proporcione los datos socio-económicos y geográficos que se le agregarán posteriormente a dicho mapa.

### **3.3 Requisitos Funcionales (RF)**

El producto contará con un conjunto de funciones necesarias para la representación, modelación y análisis de la información geográfica en la Universidad de las Ciencias Informáticas. Los usuarios que emplearán el mismo, son los responsables del manejo

## CAPÍTULO 3: Presentación de la solución propuesta

---

---

y control de la información, accediendo sólo a las funcionalidades del sistema a las que tienen acceso.

El producto debe ser capaz de cumplir los siguientes requisitos funcionales:

- 1** El sistema debe permitir al usuario realizar diferentes acciones de navegación sobre el mapa.
  - 1.1** El sistema debe permitir al usuario acercar el mapa (zoom +).
  - 1.2** El sistema debe permitir al usuario alejar el mapa (zoom -).
  - 1.3** El sistema debe permitir al usuario mover el mapa hacia la izquierda sin modificar su escala (paneo).
  - 1.4** El sistema debe permitir al usuario mover el mapa hacia la derecha sin modificar su escala (paneo).
  - 1.5** El sistema debe permitir al usuario mover el mapa hacia arriba sin modificar su escala (paneo).
  - 1.6** El sistema debe permitir al usuario mover el mapa hacia abajo sin modificar su escala (paneo).
- 2** El sistema debe permitir que el usuario pueda seleccionar las capas que desee visualizar en el mapa.
- 3** El sistema debe permitir que el usuario pueda medir la distancia entre dos lugares del mapa.
- 4** El sistema debe permitir que el usuario realice búsquedas y localizaciones temáticas en el mapa.
  - 4.1** El sistema debe permitir que el usuario realice búsquedas y localización en el mapa de persona según el solapín, carnet de identidad o usuario.
  - 4.2** El sistema debe permitir que el usuario realice búsquedas y localización en el mapa de edificaciones según el área temática a la que pertenezca.
  - 4.3** El sistema debe permitir que el usuario realice búsquedas y localización de edificios según el número del mismo.
- 5** El sistema debe brindar información asociada al elemento que se esté buscando, ya sea personas, edificios o edificaciones.

## CAPÍTULO 3: Presentación de la solución propuesta

---

---

### 3.4 Requisitos No Funcionales (RNF)

#### Usabilidad

- La aplicación será funcional para todos los dispositivos móviles que contengan un navegador WAP.

#### Soporte

- A la aplicación se le brindará soporte y actualización en el tiempo que estime el equipo de desarrollo.

#### Software para el cliente

- Debe contar con un navegador WAP que interprete lenguaje XML.

#### Software para el Servidor

- Sistemas operativos GNU/Linux.
- Servidor Web Apache 2.x.
- PostgreSQL 8.6 como Sistema Gestor de Base de Datos.
- PostGis como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.6.6.

#### Hardware para el servidor:

- El Servidor de Mapas tenga como mínimo 1GB de RAM y 40GB de disco duro.
- El Servidor de base de datos debe tener como mínimo 1GB de RAM y 40GB de disco duro.
- Procesador 3 GHz como mínimo.
- Debe contar con tarjeta de red.

#### Requisitos de Licencia

- De acuerdo a los tipos de licencias de los componentes y herramientas que se proponen a utilizar se puede catalogar legalmente esta aplicación de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

#### Requisitos Legales, Derecho de Autor y otros.

- La mayoría de las herramientas de desarrollo son libres y del resto las licencias están avaladas.

## CAPÍTULO 3: Presentación de la solución propuesta

---

---

- La aplicación se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

### 3.5 Descripción del Sistema Propuesto

La propuesta de solución en el modelado de análisis de la aplicación para dispositivos móviles contemplará las opciones del SIGUCI para conocer sus características e información socio-económica de sus entidades. La aplicación constará además con las funcionalidades básicas de los SIG (zoom, paneo, centrar mapa y selección de capas), medición de distancia, búsquedas temáticas y mostrar información sobre un objeto del mapa.

#### 3.5.1 Descripción de los actores

Un actor no es parte del sistema en desarrollo, es un agente externo que interactúa con el mismo en pos de obtener un resultado esperado. El sistema cuenta con los actores que se especifican a continuación:

Actor	Descripción
Usuario	Individuo que se conecta a través de un dispositivo móvil y utiliza las funcionalidades de la aplicación.

Tabla 1 Actor del sistema

### 3.5.2 Diagrama de Casos de Uso del Sistema

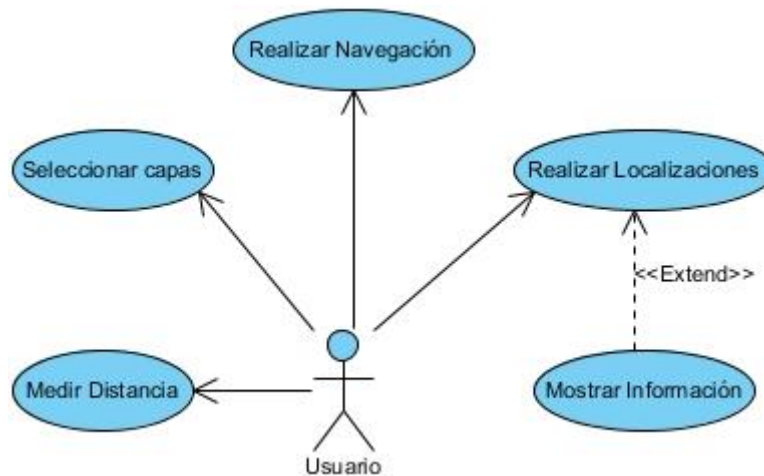


Ilustración 3 Diagrama de casos de usos del sistema

### 3.5.3 Descripción del Diagrama de Caso de Uso

El sistema cuenta con cinco casos de uso, de los cuales tres estaban en la aplicación con anterioridad, como es el caso de Seleccionar Capas, Realizar Navegación y Realizar Localizaciones. Los casos de uso Medir Distancia y Mostrar Información fueron agregados como nuevas funcionalidades al SIGUCI para dispositivos móviles.

El caso de uso Medir Distancia hace referencia al RF 3, pretende que el usuario pueda conocer la distancia estimada entre dos lugares del mapa, estos lugares pueden ser dos edificios, dos puntos de interés o entre un edificio y un punto de interés. El caso de uso inicia cuando el usuario selecciona la opción de medir distancia y marca dos objetivos en el mapa. Termina cuando el sistema le muestra la distancia estimada en metros. Para conocer más información asociada a este caso de uso se puede consultar la descripción textual ubicada en los anexos de este trabajo.

El caso de uso Mostrar Información hace referencia al RF 5, el mismo tiene la cualidad de ser extendido de Realizar Localizaciones, pretende que el usuario pueda obtener información referente a un objetivo del mapa una vez que se halla localizado con el caso de uso Realizar Localizaciones. El caso de uso inicia cuando el usuario selecciona la opción de “ver detalles” del objetivo señalado del mapa después de realizada una localización y termina cuando el sistema muestra la información del objetivo. Para conocer más información asociada a este caso de uso se puede consultar la descripción textual ubicada en los anexos de este trabajo.



### **3.6 Conclusiones**

En este capítulo se explicó la propuesta de solución del análisis para la aplicación informática, se decidió realizar la modelación a través de un modelo de dominio ya que no se identificaron procesos de negocios. Con el levantamiento de requisitos se identificaron catorce requisitos funcionales y veinte y tres requisitos no funcionales. Una vez establecidos los requisitos funcionales se identificó el actor del sistema (usuario) y los seis casos de usos referentes a dichos requisitos, los cuales fueron plasmados en el diagrama de casos de uso del sistema. Mediante la elaboración de las descripciones textuales de los casos de uso se facilitó el entendimiento de cómo trabajará el sistema. Una vez concluida esta etapa del análisis se puede comenzar la propuesta del diseño para la aplicación informática.

### *Capítulo IV*

## *Construcción de la solución propuesta.*

### **4.1 Introducción**

En este capítulo se realizará la construcción de la aplicación diseñada en el capítulo anterior. Para elaborarla se identificarán los patrones de diseño a emplear y los principales artefactos del diseño para la solución propuesta como el diagrama de clases del diseño para el caso de uso Medir Distancia, se mostrará además el modelo de datos, el modelo de despliegue y el modelo de implementación. Una vez finalizada la aplicación y realizada las pruebas de Caja Negra se mostrarán sus resultados.

### **4.2 Patrón de arquitectura de *software***

Los patrones de arquitectura de *software* son una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural para un sistema de *software*, que consta de subsistemas, sus responsabilidades e interrelaciones. Uno de los aspectos más importantes de los patrones arquitectónicos es que representan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad.

#### **4.2.1 Patrón modelo-vista-controlador (MVC)**

El MVC es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica del control en tres componentes distintos. Un modelo puede tener diversas vistas, cada una con su correspondiente controlador.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema).
- Llevar un registro de las vistas y controladores del sistema.

## CAPÍTULO 4: Construcción de la solución propuesta.

---

- Si se está ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero *bath* que actualiza los datos, un temporizador que desencadena una inserción, etc.). (37)

El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contener las reglas de gestión de eventos. (37)

Las vistas son responsables de:

- Recibir datos del modelo y la muestra al usuario.
- Tener un registro de su controlador asociado. (37)

Para la aplicación se escoge este patrón de arquitectura debido a que este trabajo de diploma sigue como base el proyecto de investigación y desarrollo de SIGUCI para celulares y tienen como patrón de diseño el MVC, y por ende no existiría incompatibilidad.

### 4.3 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*. Brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Están compuestos por un nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

#### 4.3.1 Patrones generales de *software* para asignación de responsabilidades. Patrones GRASP

Estos patrones describen los principios fundamentales de diseño para la asignación de responsabilidades. Aplican el razonamiento para el diseño de una forma sistemática, racional y explicable. En la construcción del diseño e implementación del *software* de este trabajo se definió utilizar los siguientes patrones GRASP:

**Patrón Bajo Acoplamiento:** transmite la idea de tener las clases lo menos relacionadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de

## CAPÍTULO 4: Construcción de la solución propuesta.

---

---

las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. (38)

**Patrón Alta Cohesión:** pretende que la información almacenada en una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Las clases tienen responsabilidades moderadas y colabora con otras clases para llevar a cabo las tareas. Una clase con alta cohesión posee un número relativamente pequeño de métodos, con funcionalidad altamente relacionada. Si la tarea es extensa se apoya en otros objetos. Es ventajosa porque es relativamente fácil de mantener, entender y reutilizar. (38)

**Patrón Experto:** es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada. Como beneficios de este patrón se tiene que se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas, se distribuye el comportamiento entre las clases que contienen la información requerida y por ende son más fáciles de entender y mantener. (38)

**Patrón Creador:** identifica quién debe ser el responsable de la creación de nuevos objetos o clases. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización. (38)

**Patrón Controlador:** se utiliza como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, todo esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento. (38)

### 4.4 Modelo de diseño

El modelo de diseño ayuda a entender, aclarar y transmitir las ideas que tiene sobre su código y los requisitos del usuario que el sistema de *software* debe satisfacer. Describe como se relacionan y comportan los casos de uso y las estructuras de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Los diagramas de clases de diseño son utilizados durante el proceso del diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre ellos. Entre los artefactos que genera el modelo de diseño se encuentra el Diagrama de Clases del Diseño donde se muestran las clases, interfaces y colaboraciones, así como las relaciones entre ellas.

## 4.4.1 Diagrama de clases del diseño para el Caso de Uso Medir Distancia

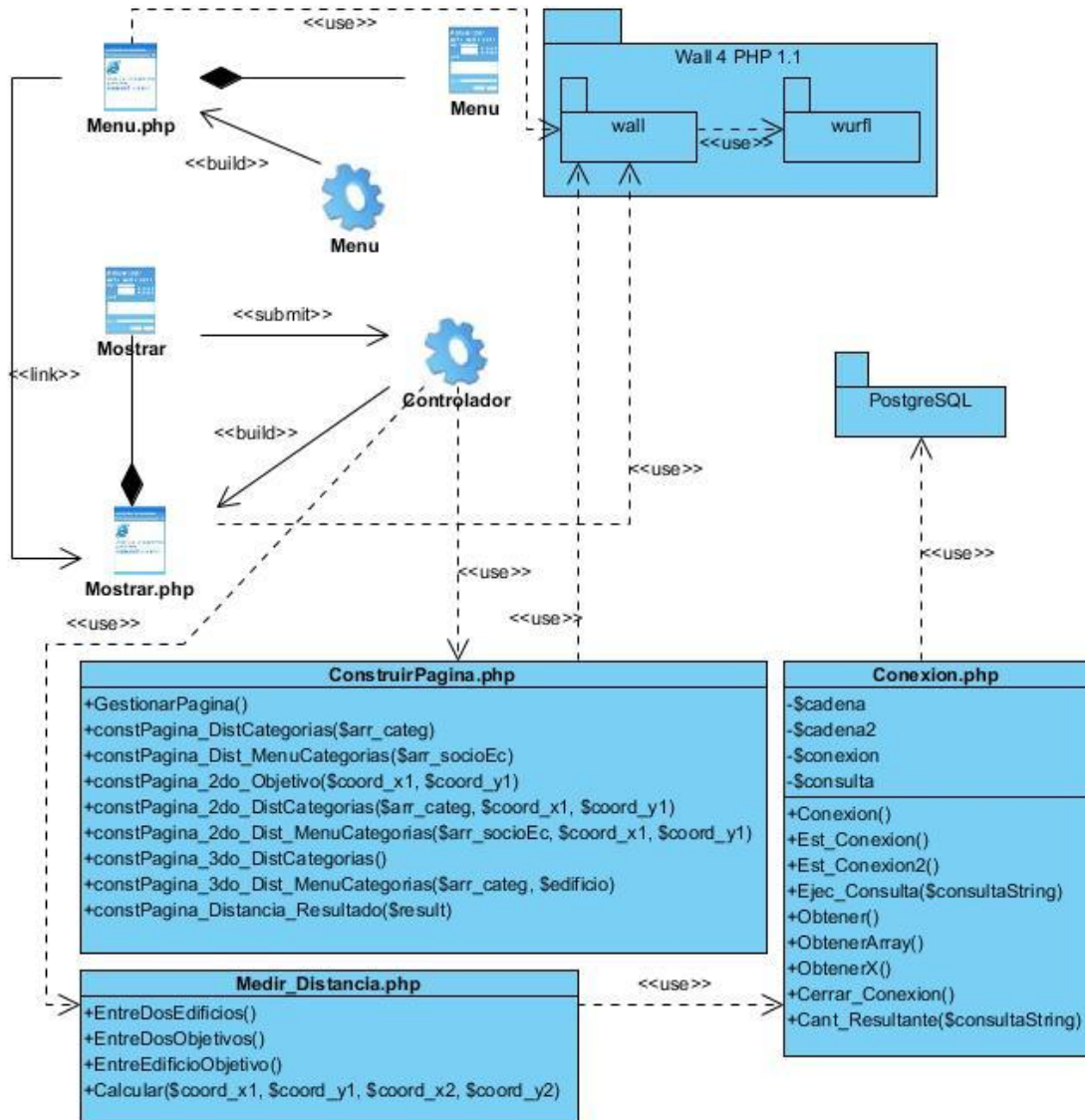


Ilustración 4 Diagrama de clase del diseño del Caso de Uso Medir Distancia

## 4.5 Modelo de datos

Un modelo de datos permite describir los elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí. Es un lenguaje utilizado para la descripción de una base de datos permitiendo describir las estructuras de datos de la base.

# CAPÍTULO 4: Construcción de la solución propuesta.

## 4.5.1 Diagrama de clases persistentes

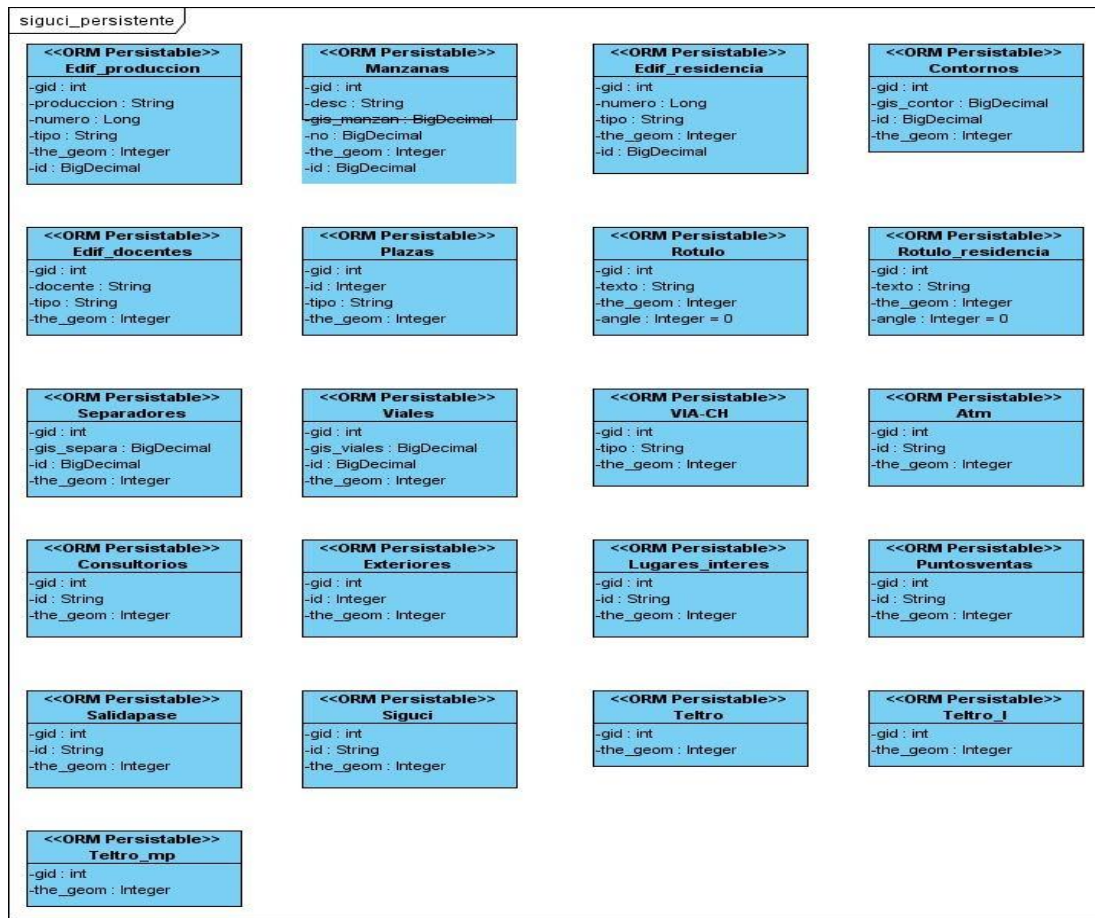


Ilustración 5 Clases persistentes del modelo entidad-relación

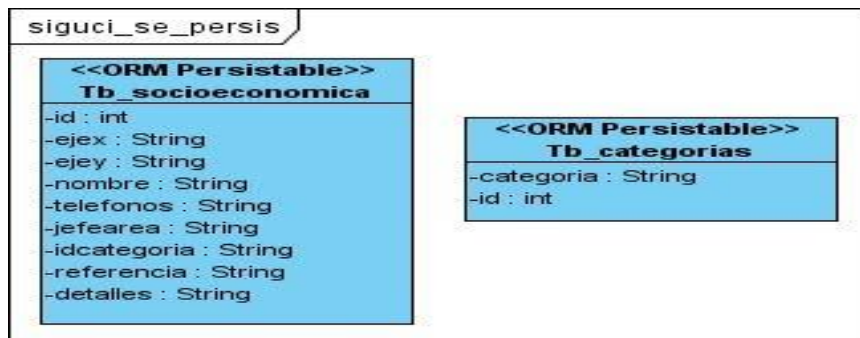


Ilustración 6 Clases persistentes de la información socio-económica

# CAPÍTULO 4: Construcción de la solución propuesta.

## 4.5.2 Modelo Entidad-Relación

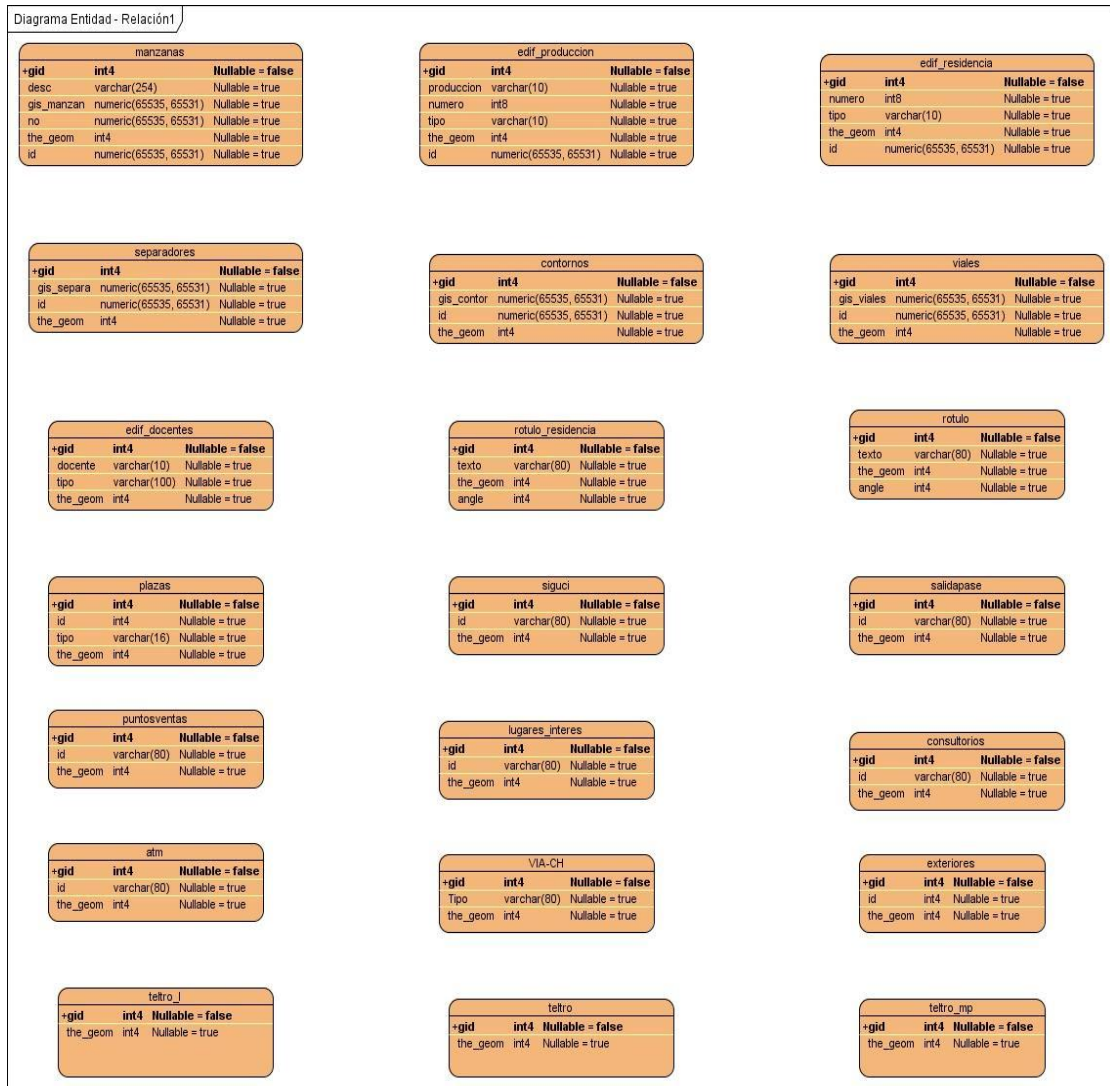


Ilustración 7 Modelo entidad-relación de la base de datos

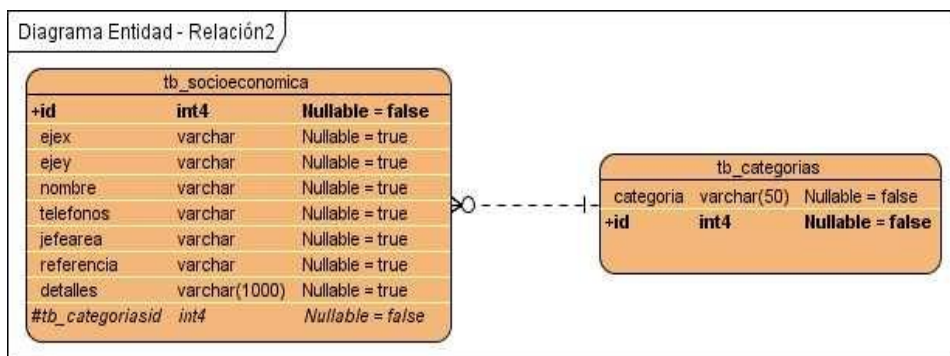


Ilustración 8 Tabla de relación socio-económica



## CAPÍTULO 4: Construcción de la solución propuesta.

### 4.5.3 Descripción de las tablas **tb\_socioeconómica** y **tb\_categorías** de la base de datos

Entre las tablas más importantes se encuentran las descritas a continuación, las mismas almacenan toda la información socio-económica conjunto a la ubicación geográfica de las distintas áreas de la universidad.

<b>Nombre</b>	tb_socioeconómica	
<b>Descripción</b>	Esta tabla guarda la información socio-económica acerca de las distintas áreas de interés de la universidad.	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
<b>id</b>	<b>int</b>	La llave primaria de la tabla.
<b>ejex</b>	<b>varchar</b>	Coordenada X donde se encuentra la localización del área.
<b>ejeY</b>	<b>varchar</b>	Coordenada Y donde se encuentra la localización del área.
<b>nombre</b>	<b>varchar</b>	Nombre del área.
<b>teléfonos</b>	<b>varchar</b>	Teléfono del área
<b>Jefeárea</b>	<b>varchar</b>	Jefe de área
<b>Idcategoría</b>	<b>varchar</b>	Llave foránea, que permite comunicar con la tabla categoría.
<b>referencia</b>	<b>varchar</b>	Dato más entendible de donde se encuentra el objetivo ejemplo: frente a la plaza mella.
<b>detalles</b>	<b>varchar</b>	Datos acerca del área. Ejemplo: el Docente 1, conocido como el "Docente Verde" se inauguró en octubre del año 2003.

Tabla 2 Descripción de la tabla **tb\_socioeconómica** de la base de datos

<b>Nombre</b>	tb_categorías	
<b>Descripción</b>	Tabla que contiene todas las áreas de la universidad.	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
<b>categoría</b>	<b>varchar</b>	Nombre del área.
<b>id</b>	<b>int</b>	Llave primaria de la tabla.

Tabla 3 Descripción de la tabla **tb\_categoria** de la base de datos

### 4.6 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros.

#### 4.6.1 Diagrama de Componente

El diagrama de componente representa cómo un sistema de *software* es dividido en elementos y muestra las dependencias entre estos elementos. Los elementos físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes.

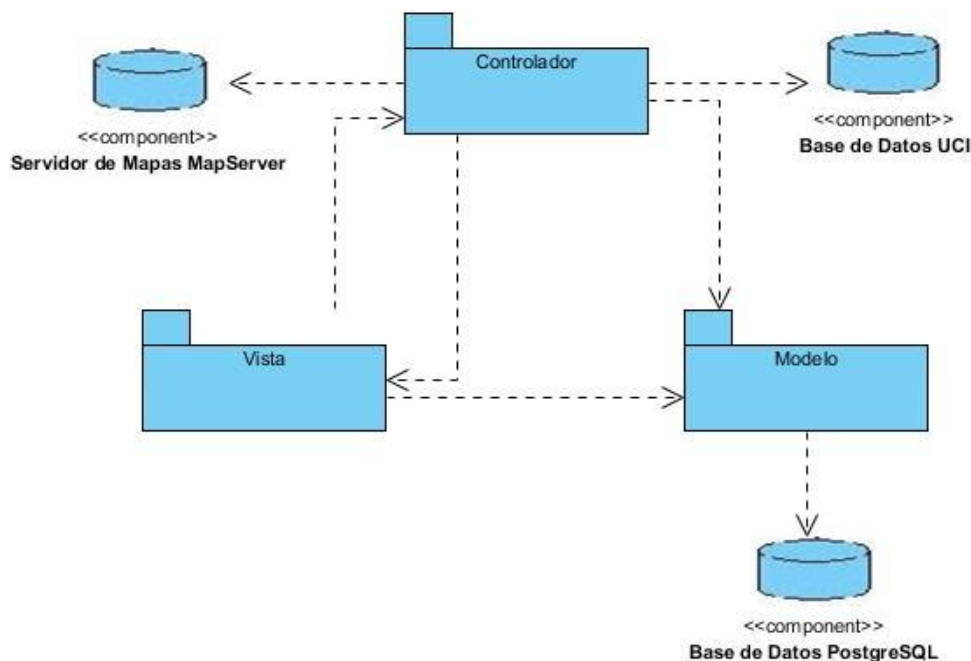


Ilustración 9 Vista general del diagrama de componentes

Para comprender y detallar más específico este diagrama se expone una vista detallada de los paquetes Modelo, Vista, Controlador y los componentes que contiene cada paquete en particular.

## CAPÍTULO 4: Construcción de la solución propuesta.

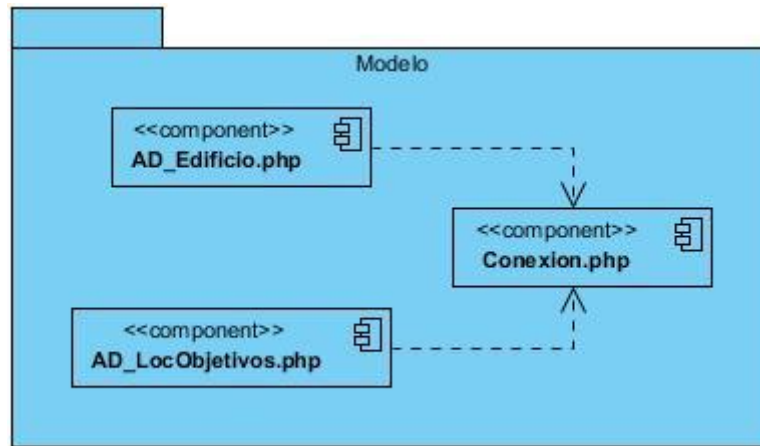


Ilustración 10 Vista detallada del Modelo

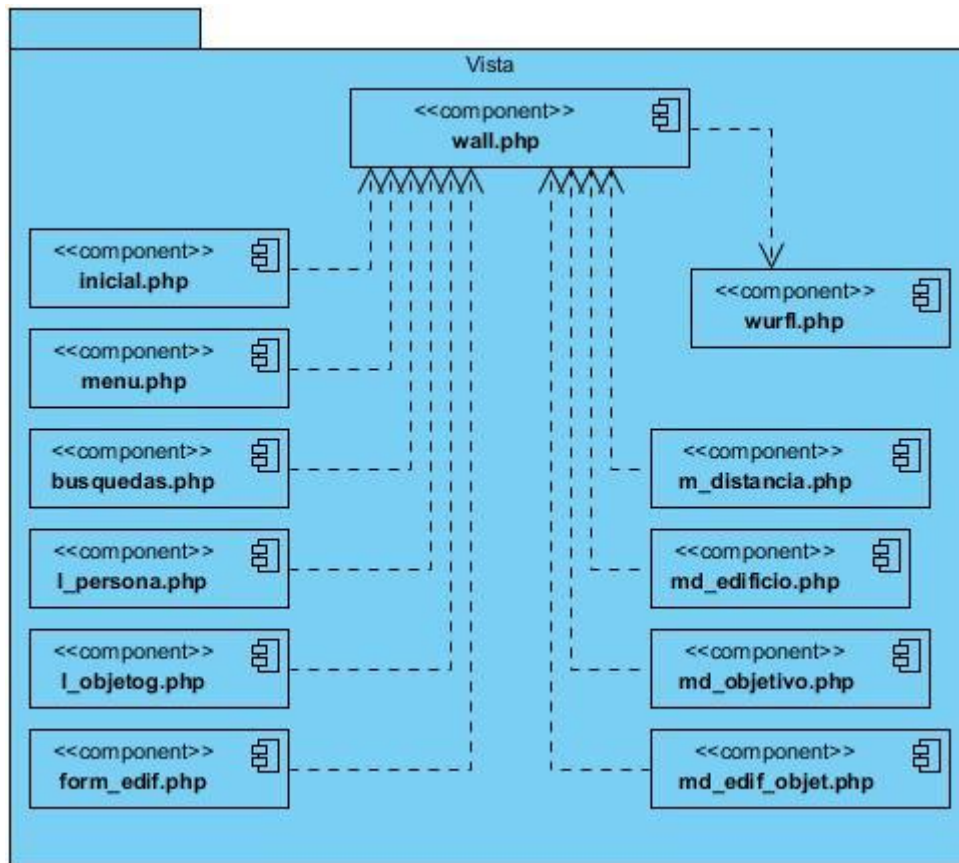


Ilustración 11 Vista detallada de la Vista

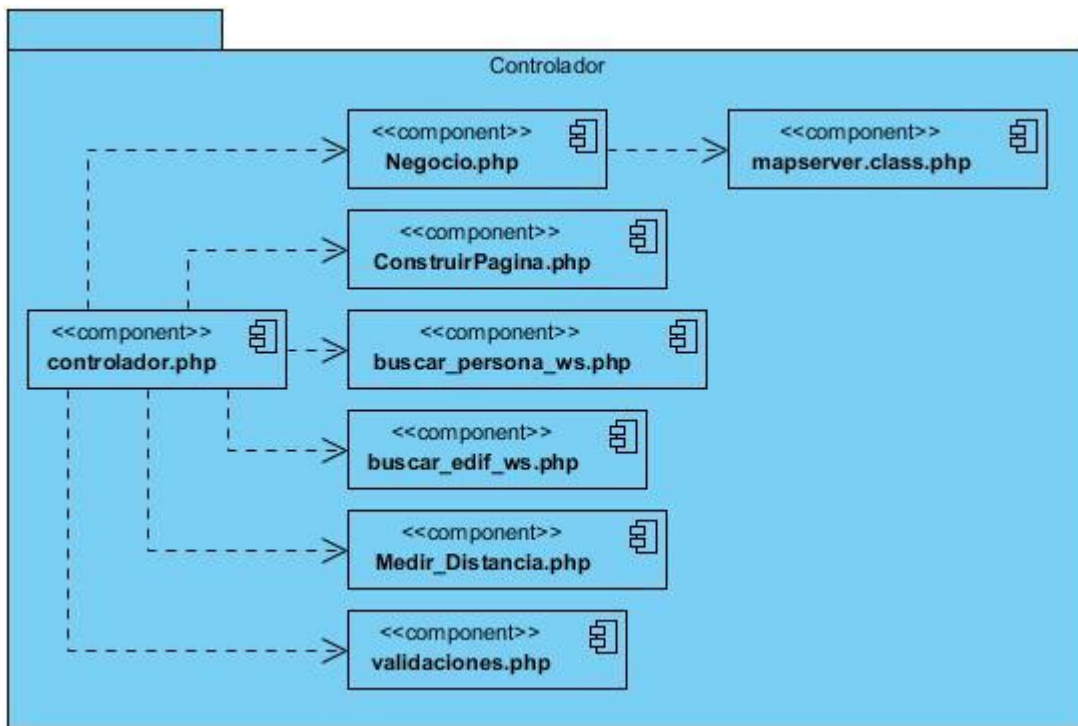


Ilustración 12 Vista detallada del Controlador

### 4.7 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de *hardware* similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos.

## CAPÍTULO 4: Construcción de la solución propuesta.

---

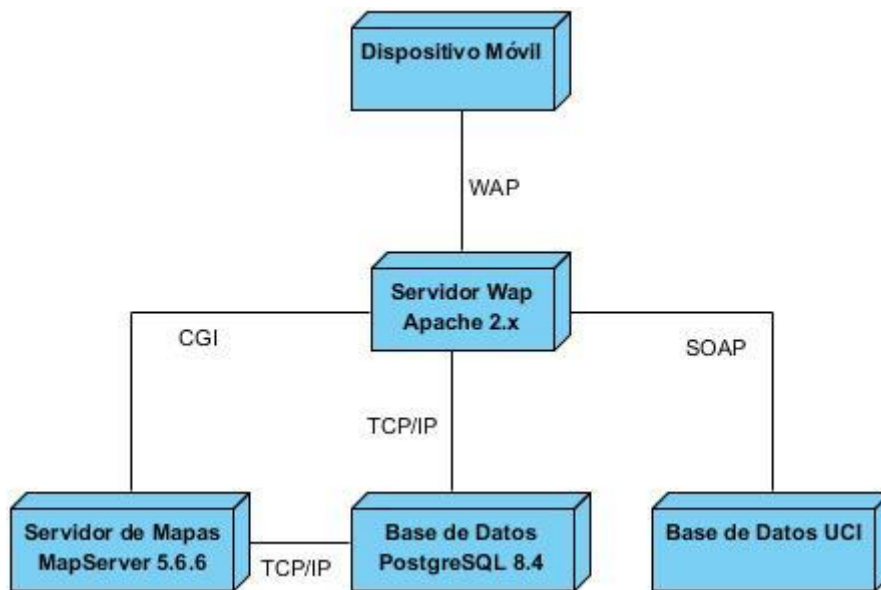


Ilustración 13 Modelo de despliegue

En este modelo se muestra como el dispositivo móvil se conecta a través del protocolo Wap al servidor Wap Apache 2.x donde se encuentra la aplicación. Este servidor está conectado con el servidor de mapas MapServer 5.6.6 por CGI (*Common Gateway Interface*), al servidor de base de datos PostgreSQL 8.4 con su extensión PostGIS mediante TCP/IP (*Transmission Control Protocol / Internet Protocol*) y al servidor de Bases de Datos UCI de la universidad mediante SOAP (*Simple Object Access Protocol*) para la manipulación de datos referentes a personas y edificaciones.

El servidor de mapas MapServer 5.6.6 es el encargado de administrar los mapas y las capas, para hacerlo se auxilia de una serie de consultas realizadas al servidor de base de datos PostgreSQL 8.4 el cual es el encargado de manejar toda la información sobre la cartografía utilizada en la aplicación.

### 4.8 Diseño de pruebas al sistema

Realizar pruebas a un sistema es imprescindible para verificar la calidad y el adecuado funcionamiento del mismo. Este proceso se lleva a cabo con el objetivo de comprobar que el sistema satisface los requisitos y se comporta de acuerdo a lo establecido.

#### 4.8.1 Pruebas de caja negra

Esta técnica se realiza sobre la interfaz del programa a probar, verificando desde la interfaz, las entradas y las salidas de dicho programa. Para llevar a cabo esta técnica

## CAPÍTULO 4: Construcción de la solución propuesta.

---

---

no se necesita conocer la lógica interna de un programa, únicamente se verifica que la funcionalidad cumpla su objetivo.

Para la realización de las pruebas de caja negra se escoge métodos: basados en grafos y las técnicas: partición equivalente y análisis de valores límite. Se le realizó pruebas a los caso de usos Medir Distancia y Mostrar Información. Una vez realizadas las pruebas dieron como resultado la existencia de algunos errores que fueron subsanados y en un segundo despliegue de las pruebas se demostró que no existían otros errores. Los artefactos relacionados a la aplicación de estas pruebas de adjuntan como anexo de este trabajo.

### **4.9 Conclusiones**

Una vez concluido este capítulo, está completamente diseñado y construido el sistema. Se hizo uso del modelo de arquitectura MVC siendo de gran ayuda por su organización para agrupar los componentes, se definieron los patrones GRASP para realizar el diseño de la aplicación. Con la modelación de las clases del diseño se logró entender el funcionamiento de la aplicación permitiendo su implementación. Quedaron definidas las clases persistentes, siendo estas las clases creadas para describir las estructuras de datos de la base del modelo entidad-relación. Con la modelación del diagrama de componente se observa la estructura de los ficheros dentro de cada paquete cumpliendo siempre con la arquitectura MVC. Se definió el diagrama de despliegue, quedando definida la distribución del *hardware* involucrado para la utilización de la aplicación. Una vez finalizada la aplicación se le realizó las pruebas de Caja Negra al sistema, donde finalmente se subsanaron los errores encontrados.

### *Conclusiones generales*

Con la realización de este trabajo, se considera cumplido el objetivo que dio inicio al desarrollo de esta investigación. Haciendo un uso correcto de los métodos científicos que se pusieron en práctica, se logró estudiar el objeto de estudio, así como el entorno que rodeaba al problema en cuestión. A lo largo de la investigación se fueron dando cumplimiento a las tareas investigativas trazadas al comienzo de la misma, se fundamentó en el estudio de los SIG y dispositivos móviles, se identificaron los principales servicios Web estandarizados de la OGC, se generó una documentación técnica asociada al sistema y a los procesos que se llevaron a cabo durante su desarrollo a través de los artefactos generados a partir de la puesta en práctica de la metodología RUP. Todo esto logró un mayor entendimiento del sistema y la posibilidad de perfeccionar una idea que pudiera ser retomada en un futuro.

Como punto culminante de esta investigación, se logró el desarrollo de un Sistema de Información Geográfica en dispositivos móviles para la Universidad de las Ciencias Informáticas capaz de ubicar geográficamente el mapa de la universidad y realizar trabajos sobre el mismo, perfeccionando de esta forma la aplicación existente convirtiéndola en una herramienta más sólida y funcional. Se les dio uso a los servicios Web estandarizados de la OGC, usando para el desarrollo de la aplicación el WMS. Se insistió en todo momento en la utilización de *software* y herramientas libres o con las licencias avaladas atendiendo a la política de independencia y libertad de *software*. El *software* cumple con los requisitos planteados para su desarrollo por lo que se convierte en una herramienta capaz de brindar los resultados esperados.

### *Recomendaciones*

La existencia de un Sistema de Información Geográfica de la universidad para dispositivos móviles es de gran ayuda para la búsqueda y localización de personas y edificaciones, las metas planteadas con este trabajo se han cumplido y en base a los resultados obtenidos se recomienda:

- ✓ Realizar pruebas de Caja Blanca a la aplicación para validar su código.
- ✓ Poner en funcionamiento la aplicación en la red universitaria.
- ✓ Agregar nuevas funcionalidades a la aplicación como el cálculo del camino mínimo.



## Anexos

### Caso de Uso Medir Distancia

<b>Caso de Uso:</b>	Medir Distancia
<b>Actor:</b>	Usuario
<b>Propósito:</b>	Con esta funcionalidad se pretende que el usuario pueda conocer la distancia estimada entre dos objetivos del mapa.
<b>Resumen:</b>	El caso de uso inicia cuando el usuario selecciona la opción de medir distancia y marca dos objetivos en el mapa. Termina cuando el sistema le muestra la distancia estimada en metros.
<b>Precondiciones:</b>	El sistema debe mostrar la opción de medir distancia.
<b>Referencias</b>	RF 3
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor selecciona la opción Medir Distancia.	2. El sistema muestra tres opciones, “entre dos edificios”, “entre dos puntos de interés” o “entre un edificio y un punto de interés”. (ver Interfaz 1)
3. El usuario selecciona una de las tres opciones.	4. El sistema realiza la operación según la selección del usuario:  Si el usuario selecciona “entre dos edificios” se muestra un formulario dando la posibilidad al usuario de

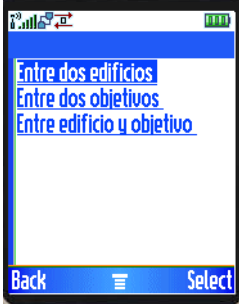
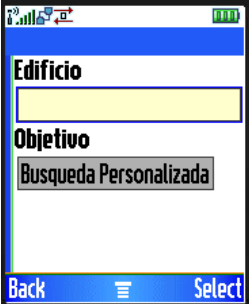
	<p>entrar los números de ambos edificios.</p> <p>Si el usuario selecciona “entre dos puntos de interés” se muestran dos listas con todas las estructuras para que sean seleccionadas por el usuario.</p> <p>Si el usuario selecciona “entre un edificio y un punto de interés” se muestra un formulario dando la posibilidad al usuario de entrar el número del edificio y una lista para seleccionar el punto de interés. (ver Interfaz 2)</p>
<p>5. El usuario llena los datos del formulario y da en calcular.</p>	<p>6. El caso de uso termina cuando el sistema muestra al usuario la información de distancia expresada en metros.</p>
<p><b>Prototipo de Interfaz</b></p>	
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <p><b>Interfaz 1</b></p> </div> <div style="text-align: center;">  <p><b>Interfaz 2</b></p> </div> </div>	
<p><b>Poscondiciones:</b></p>	<p>El sistema visualiza en el móvil un número que representa la distancia entre estos los dos objetivos.</p>

Tabla 4 CU Medir Distancia

### Caso de Uso Mostrar Información

<b>Caso de Uso:</b>	Mostrar Información	
<b>Actor:</b>	Usuario	
<b>Propósito:</b>	Con esta funcionalidad se quiere que el usuario pueda obtener información referente a un objetivo del mapa.	
<b>Resumen:</b>	El caso de uso inicia cuando el usuario selecciona la opción de “ver detalles” del objetivo señalado del mapa después de realizada una localización y termina cuando el sistema muestra la información del objetivo.	
<b>Precondiciones:</b>	El sistema tiene que haber realizado una localización y tener activo el objetivo señalado.	
<b>Referencias:</b>	RF 5	
<b>Prioridad:</b>	Secundario	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El caso de uso inicia cuando el actor solicita la opción “ver detalles” de un objeto señalado en el mapa después de una localización. (ver Interfaz 3)	2. El sistema busca en la base de datos la información socio-económica asociada al objetivo.	
	3. El caso de uso termina cuando el sistema muestra la información del objeto seleccionado. (ver Interfaz 4)	


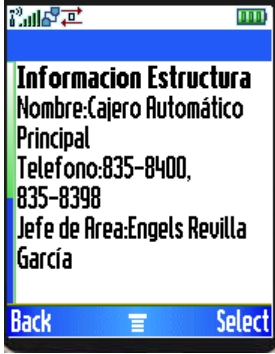
Prototipo de Interfaz	
	
<p><b>Interfaz 3</b></p>	<p><b>Interfaz 4</b></p>
<b>Poscondiciones:</b>	El sistema visualiza la información del objetivo.

Tabla 5 CU Mostrar Información

Diagrama de clases del diseño para el Caso de Uso Mostrar Información

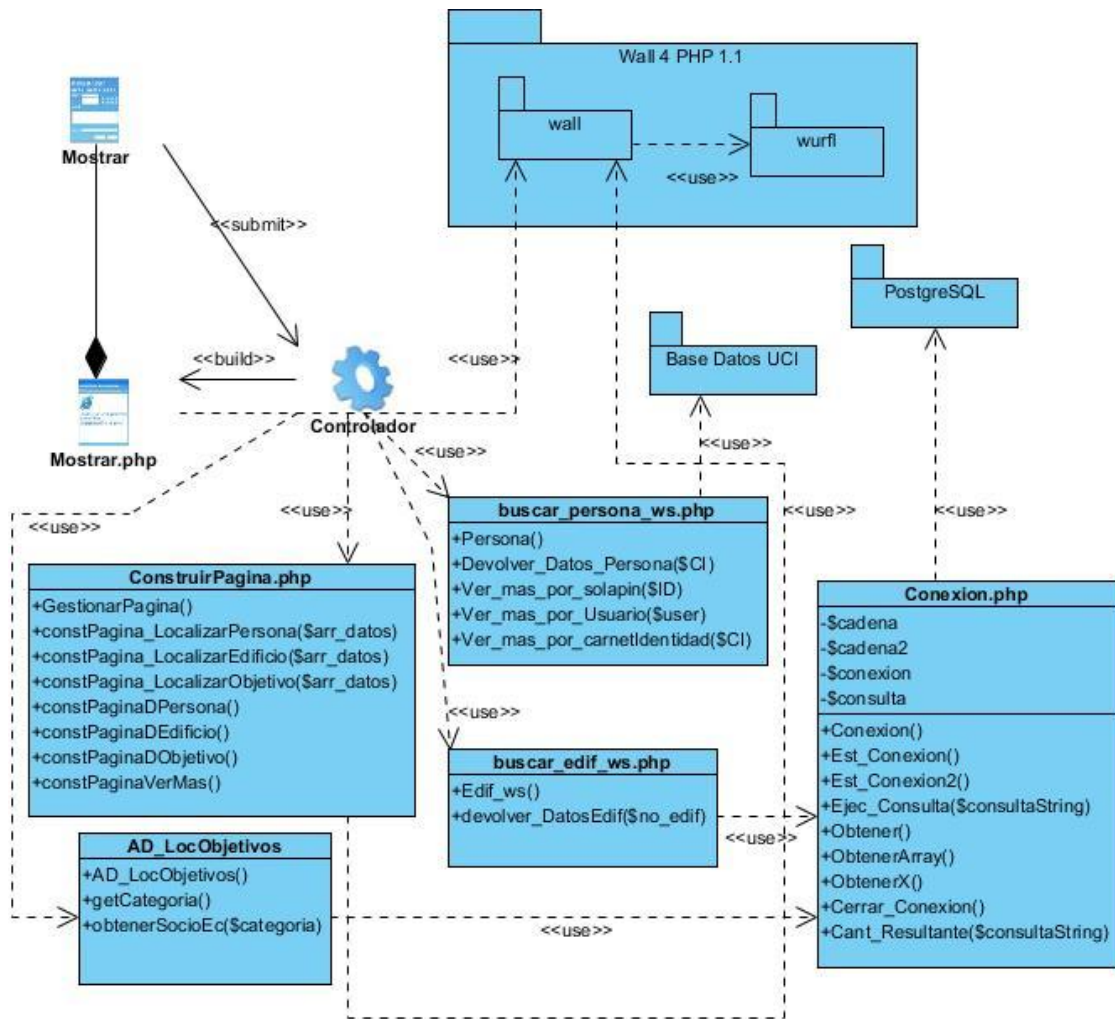


Ilustración 14 Diagrama de clase del diseño del Caso de Uso Mostrar Información

## Caso de prueba de Caja Negra del CU Medir Distancia

**Descripción general:** el caso de uso inicia cuando el usuario selecciona la opción de medir distancia y marca dos objetivos en el mapa. Termina cuando el sistema le muestra la distancia estimada en metros.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: medir distancia entre dos edificios.	EC 1. Insertar datos de forma correcta.	Esta funcionalidad permite realizar un cálculo estimado de la distancia en metros entre dos edificios de residencia en la universidad.	<ul style="list-style-type: none"> <li>• Página Inicial.</li> <li>• Clic en el botón "Menú".</li> <li>• Clic en la opción "Medir Distancia".</li> <li>• Clic en la opción "Entre dos edificios".               <ul style="list-style-type: none"> <li>• Se muestra un formulario con dos textbox y un botón "Calcular".</li> </ul> </li> <li>• Llenar los campos con números entre el 1 y el 150.</li> <li>• Clic en el botón "Calcular".</li> <li>• Se muestra un formulario con el resultado de la distancia entre ambos edificios.</li> <li>• Clic en el botón "Volver".</li> </ul>

	EC 1.2. Insertar datos de forma incorrecta.	El sistema no calcula la distancia debido a que hay un error en el valor insertado o se ha dejado el campo vacío.	<ul style="list-style-type: none"> <li>• Página Inicial.</li> <li>• Clic en el botón “Menú”.</li> <li>• Clic en la opción “Medir Distancia”.</li> <li>• Clic en la opción “Entre dos edificios”.</li> <li>• Se muestra un formulario con dos textbox y un botón “Calcular”.</li> <li>• Dejar un textbox vacío o entrar el dato incorrecto. <ul style="list-style-type: none"> <li>• Clic en el botón “Calcular”.</li> </ul> </li> <li>• Muestra un formulario con el error.</li> </ul>
--	---	---	--

Tabla 6 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Medir Distancia

A partir de esta descripción se detallan las variables que se encuentran en toda la interfaz asociadas al caso de uso que se le estaba diseñando el caso de prueba. Esta descripción está compuesta por campos tales como:

- **No:** se enumera todos los campos o variable, descrito en el caso de uso.
- **Nombre de campo:** se especifica el nombre del campo de entrada.
- **Clasificación:** se especifica la clasificación según el componente de diseño utilizado (ejemplo: campo de texto, botón).
- **Puede ser nulo:** se especifica si el campo puede ser nulo o no, para ello solo se puso Sí o No.
- **Descripción:** se describen brevemente los datos que debían introducirse.

**Descripción de Variables:**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Números de los edificios.	Campo de Texto.	NO	Debe introducir los números de los dos edificios que se le van a medir la distancia.

**Tabla 7 Descripción de Variable del Caso de Uso Medir Distancia**

Esta descripción posibilita que se ejecute una matriz de datos, donde se evalúa y se prueba la validez de cada uno de los datos introducidos en el sistema, específicamente en la sección que se estará probando. Utilizando un juego de datos válidos e inválidos se identificará el empleo de la técnica de partición de equivalencia y análisis de valores límites.

Esta matriz de datos contiene los siguientes aspectos:

- **Escenario:** se especifica el nombre del escenario.
- **Variables [1, 2,..., n]:** se especifica el nombre de la variable, o el número según la tabla de descripción de variables y en su celda correspondiente se indicó el valor del dato [V (Válido), I (Inválido), N/A (No Aplica)].
- **Respuesta del sistema:** se escribe el resultado que se esperaba al realizar la prueba.
- **Resultado de la prueba:** se escribe el resultado que se obtuvo al realizar la prueba. (Satisfactorio o No Satisfactorio).



**Matriz de dato:**

ID del escenario	Escenario	V1 Opciones de medición.	V2 Edificio 1	V5 Edificio 2	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Medir distancia con datos correctos.	V/dos edificios	V/150	V/112	El sistema muestra la distancia expresada en metros.	Satisfactoria.
		V/dos edificios	V/104	V/108		Satisfactorio.
EC 1.2	Medir distancia con datos incorrectos	V/dos edificios	I/160	I/155	El sistema muestra un formulario con los errores encontrados.	Satisfactorio
		V/dos edificios	I/-50	I/Ab		Satisfactorio.

**Tabla 8 Matriz de Datos Medir Distancia**

### Caso de prueba de Caja Negra del CU Mostrar Información

**Descripción general:** el caso de uso inicia cuando el usuario selecciona la opción de “ver detalles” del objetivo señalado del mapa después de realizada una localización y termina cuando el sistema muestra la información del objetivo.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: mostrar información de un objetivo del mapa.	EC 1. Insertar datos de forma correcta.	El sistema permite mostrar los datos de una persona, edificio o punto de interés, después de haberlo buscado con la funcionalidad de Realizar Localización.	<ul style="list-style-type: none"> <li>• Página Inicial.</li> <li>• Clic en el botón “Menú”.</li> <li>• Clic en la opción “Localizaciones”.</li> <li>• Clic en la opción “Localizar Persona”.</li> <li>• Se muestra un formulario la opción de seleccionar el criterio de búsqueda, un textbox y un botón “Localizar”.</li> <li>• Llenar los campos.</li> <li>• Clic en el botón “Localizar”.</li> <li>• Se muestra un formulario informando que encontró el objetivo y dos botones, uno de “Localizar” y otro de “Ver detalles”.</li> <li>• Se selecciona el botón “Ver detalles”.</li> <li>• Se muestra la información socio-económica asociada a dicho objetivo.</li> <li>• Clic en el botón “Volver”.</li> </ul>

	EC 1.2. Insertar datos de forma incorrecta.	El sistema no muestra los datos de una persona, edificio o punto de interés, debido a que se insertó mal los datos en la funcionalidad de Realizar Localización.	<ul style="list-style-type: none"> <li>• Página Inicial.</li> <li>• Clic en el botón “Menú”.</li> <li>• Clic en la opción “Localizaciones “.</li> <li>• Clic en la opción “Localizar Edificio”.</li> <li>• Se muestra un formulario con un textbox y un botón “Localizar”.</li> <li>• Dejar un textbox vacío o entrar el dato incorrecto. <ul style="list-style-type: none"> <li>• Clic en el botón “Localizar”.</li> </ul> </li> <li>• Muestra un formulario con el error.</li> </ul>
--	---	--	--

Tabla 9 Diseño de Caso de Prueba de Caja Negra del Caso de Uso: Medir Distancia

A partir de esta descripción se detallan las variables que se encuentran en toda la interfaz asociadas al caso de uso que se le estaba diseñando el caso de prueba. Esta descripción está compuesta por campos tales como:

- **No:** se enumera todos los campos o variable, descrito en el caso de uso.
- **Nombre de campo:** se especifica el nombre del campo de entrada.
- **Clasificación:** se especifica la clasificación según el componente de diseño utilizado (ejemplo: campo de texto, botón).
- **Puede ser nulo:** se especifica si el campo puede ser nulo o no, para ello solo se puso Sí o No.
- **Descripción:** se describen brevemente los datos que debían introducirse.

**Descripción de Variables:**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1		Botón	NO	Debe presionar el botón para mostrar la información socio-económica asociada.

**Tabla 10 Descripción de Variable del Caso de Uso Medir Distancia**

Esta descripción posibilita que se ejecute una matriz de datos, donde se evalúa y se prueba la validez de cada uno de los datos introducidos en el sistema, específicamente en la sección que se estará probando. Utilizando un juego de datos válidos e inválidos se identificará el empleo de la técnica de partición de equivalencia y análisis de valores límites.

Esta matriz de datos contiene los siguientes aspectos:

- **Escenario:** se especifica el nombre del escenario.
- **Variables [1, 2,..., n]:** se especifica el nombre de la variable, o el número según la tabla de descripción de variables y en su celda correspondiente se indicó el valor del dato [V (Válido), I (Inválido), N/A (No Aplica)].
- **Respuesta del sistema:** se escribe el resultado que se esperaba al realizar la prueba.
- **Resultado de la prueba:** se escribe el resultado que se obtuvo al realizar la prueba. (Satisfactorio o No Satisfactorio).

## Matriz de dato:

ID del escenario	Escenario	V1 Opciones de medición.	V2 Edificio 1	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Mostrar Información con datos correctos.	V/una persona según su usuario.	V/dcouce	El sistema muestra la información socio-económica asociada.	Satisfactoria.
		V/una persona según su carnet de identidad.	V/88032521101		Satisfactorio.
EC 1.2	Mostrar Información con datos incorrectos.	V/una persona según su usuario	I/dddd	El sistema muestra un formulario con los errores encontrados	Satisfactorio
		V/una persona según su solapín.	I/pp		Satisfactorio.

Tabla 11 Matriz de Datos Medir Distancia

---



---

## *Bibliografías*

1. Gustavo, Buzai. *Análisis Socioespacial con Sistemas de Información Geográfica*. Buenos Aires : Lugar Editorial, 2006.
2. Alegsa. [Online] <http://www.alegsa.com.ar/Dic/dispositivo%20movil.php>.
3. Cátedra COITT. [Online] <http://catedra-coitt.euitt.upm.es/>.
4. Mobile Manufacturers. [Online] <http://www.mmfai.org/public/wireless.cfm?lang=es>.
5. Masadelante. [Online] <http://www.masadelante.com/faqs/gsm>.
6. Cátedra COITT. [Online] [http://catedra-coitt.euitt.upm.es/web\\_salud\\_medioamb/inicio.htm](http://catedra-coitt.euitt.upm.es/web_salud_medioamb/inicio.htm).
7. GSM World. [Online] <http://www.gsmworld.com/news/statistics/index.shtml>.
8. OGC Standards. OGC Making Location Count. [En línea] [Citado el: 3 de Mayo de 2012.] <http://www.opengeospatial.org/standards/>.
9. Cátedra COITT. [Online] [http://catedra-coitt.euitt.upm.es/web\\_salud\\_medioamb/inicio.htm](http://catedra-coitt.euitt.upm.es/web_salud_medioamb/inicio.htm).
10. Demystifying Information Technology and All Things Cyber. Counter Balance. [En línea] [Citado el: 27 de Febrero de 2011.] <http://www.counterbalance.org/itdemyst/itdemyst-print.html>.
11. LAS TIC Y SUS APORTACIONES A LA SOCIEDAD. TECNOLOGIA EDUCATIVA - WEB PERE MARQUÈS . [En línea] [Citado el: 27 de Febrero de 2011.] <http://peremarques.pangea.org/tic.htm>.
12. ACN (26-03-2009). «Los precios mensuales de la telefonía móvil en el estado español casi doblan los de la Unión Europea» (en catalán). Tinet. Consultado el 29-11-2009.
13. El Economista. Un test de velocidad de Internet móvil. [Online] Redacción CanalPDA, Agosto 1, 2008. [Cited: Abril 10, 2012.] <http://www.eleconomista.es/CanalPDA/2008/7471/>.
14. Ariadna González Ortiz (15-11-2008). «LTE, el salto al 4Gmóvil». Redestelecom.es. Consultado el 29-11-2009.
15. Nieto Pérez, Iván. Introducción a la tecnología WAP . El Código. [En línea] [Citado el: 1 de Febrero de 2011.] <http://www.elcodigo.net/tutoriales/wap/wap1.html>.
16. Guía Breve de Tecnologías XML. World Wide Web. [En línea] 9 de Enero de 2008. [Citado el: 31 de Enero de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.
17. W3C. World Wide Web Consortium (W3C). [En línea] 2009. [Citado el: 15 de 12 de 2011.] <http://www.w3.org/>.
18. Servidor Apache HTTP. *Massachusetts Institute of Technology*. [En línea] [Citado el: 3 de Febrero de 2011.] <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-httpd.html>.
19. Various Licenses and Comments about Them. *GNU Operating System*. [En línea] [Citado el: 3 de Febrero de 2011.] <http://www.gnu.org/licenses/license-list.html>.
20. EcuRed. [Online] <http://www.ecured.cu/index.php/RUP>.

21. Fowler, Martin. Programacionextrema. [En línea] [Citado el: 15 de Febrero de 2011.] <http://www.programacionextrema.org/articulos/newMethodology.es.html>.
22. Martin Fowler, Kendall Scott. UML Gota a Gota. 1999.
23. Visual Paradigm. [Online] <http://www.visual-paradigm.com>.
24. Grupo de Soluciones GSINNova. [En línea] [Citado el: 8 de Abril de 2012.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
25. Lutz, Mark. Learning Python, Fourth Edition. s.l.: O'Reilly Media, 2010.
26. Minutes PHP Developers Meeting. [Aut. libro] Marcus Börger, Wez Furlong y Rasmus Lerdorf. Paris: s.n., 2005.
27. NetBeans. [En línea] [Citado el: 16 de Febrero de 2011.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
28. NetBeans. [Online] [Cited: Abril 10, 2012.] [http://netbeans.org/community/releases/61/index\\_es.html](http://netbeans.org/community/releases/61/index_es.html).
29. Eclipse. [En línea] [Citado el: 19 de Febrero de 2011.] <http://www.eclipse.org/legal/epl-v10.html>.
30. PostgreSQL. [Online] <http://www.postgresql.org/about/awards>.
31. PostgreSQL. Ubuntu. [En línea] [Citado el: 19 de Febrero de 2011.] <http://www.guia-ubuntu.org/index.php?title=PostgreSQL>.
32. PostgreSQL. [Online] <http://www.postgresql.org/about/awards>.
33. GeoServer. [En línea] [Citado el: 19 de Febrero de 2011.] <http://groups.google.com/group/geoserver-es>.
34. MapServer. [En línea] [Citado el: 19 de Febrero de 2011.] <http://mapserver.org/>.
35. Matias Neiff. [Online] [Cited: Abril 8, 2012.] <http://matias.neiff.com.ar/2010/03/15/%C2%BFmapserver-vs-geoserver/>.
36. Infraestructura Colombiana de Datos Espaciales. [Online] [Cited: Abril 8, 2012.] [http://www.icde.org.co/web/ide\\_gig/blogs/-/blogs/geoserver-vs-mapserver](http://www.icde.org.co/web/ide_gig/blogs/-/blogs/geoserver-vs-mapserver).
37. Patrón "Modelo-Vista-Controlador". *Proactiva Calidad*. [En línea] [Citado el: 3 de Febrero de 2011.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
38. Mora, Roberto Canales. Informática profesional, las reglas no escritas para triunfar en la empresa. 2007.