



# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS FACULTAD 2

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

**Título:** *“Diseño e Implementación del Módulo Mapificación para el Sistema de Prevención del Delito de Venezuela”*

**Autor:**

Deylert Pérez Rivera

**Tutores:**

Ing. Juan Carlos Gómez Correa

Ing. David Gerardo Navarro Ferrer



La Habana, junio de 2013.

## DECLARACIÓN DE AUTORÍA

### DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y se reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Deylert Pérez Rivera**

---

Firma del Autor

**Juan Carlos Gómez Correa**

---

Firma del Tutor

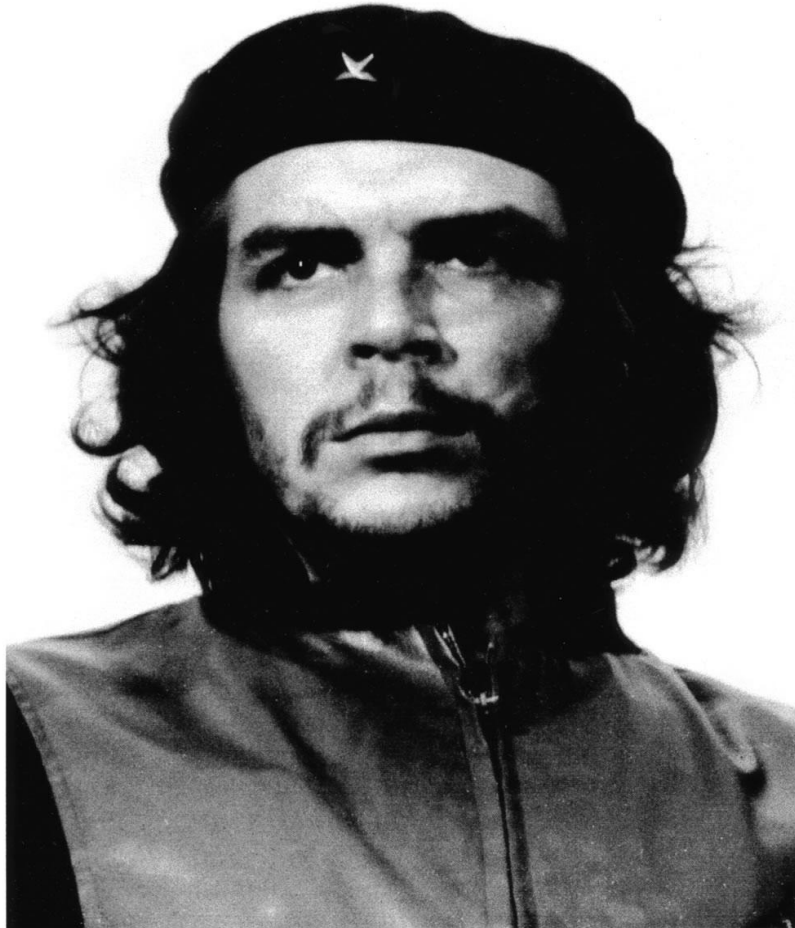
**David Gerardo Navarro Ferrer**

---

Firma del Tutor

## PENSAMIENTO

## PENSAMIENTO



*“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.”*

*Ernesto Che Guevara*

## DEDICATORIA

### DEDICATORIA

*A mis padres, José Carlos Pérez Morales y Olga Lidia Rivera Torres (E.P.D), a quienes debo todo lo que soy. A los que les estaré eternamente agradecido por haberme formado correctamente y lo mejor que pudieron, por apoyarme en todo momento, por haber confiado en mí, por enseñarme que siempre seré lo que quiera ser, por ser las personas más maravillosas del mundo. Por esto y mucho más,  
Gracias.*

*A mi abuelita querida, Naida, por ser más que mi abuela, mi madre.*

*A mis tíos, Pedro Alberto, Ñiñi, Pedro Miguel, Mirtha, Negro, Julio, Nene, René, Margarita y Xiomara por quererme tanto, por tanta dedicación, por todas las cosas lindas que a su lado aprendí, por ser fruto de su educación, porque siempre los querré y los llevaré conmigo.*

*A mis hermanos Daniel, Deilenys, Albertico, Marwin, Dilán y Lianné por estar conmigo en los buenos y malos momentos, por formar parte de mi fuente de inspiración.*

*A mi querido amor Celia, gracias por dejarme ser parte de ti, por permitirme compartir tú vida conmigo, por todo lo que tú sonrisa aporta a mis días, por estar junto a mi lado cuando lo he necesitado.*

## AGRADECIMIENTOS

### AGRADECIMIENTOS

*A las personas más importantes del mundo, mis padres, por su apoyo incondicional, lo son todo para mí y los quiero desde lo más profundo de mi corazón.*

*A mi abuelita Naida, por todo lo que me ha dado desde que nací.*

*A mis tíos Pedro Alberto, Niñi, Pedro Miguel, Mirtha, Negro, Nene, René, Margarita, Carlos, Yanet y Xiomara, por apoyarme en todo momento.*

*A mi esposa Celia por el amor que me ha brindado y por estar siempre a mi lado.*

*A mis suegros Ruso y Magalis por haberme acogido como un hijo.*

*A mis tutores Juan Carlos y David Gerardo, por todo el apoyo incondicional, por el tiempo que me dedicaron, por contribuir a mi formación como profesional, ustedes también son parte de este logro.*

*A los profesores que han contribuido a alcanzar este gran sueño, en especial a los profesores Sucel, Yoe, Liván, Maura, Tralys, Edgar y Deivis.*

*No quiero dejar pasar por alto a aquellas personas que durante los 5 años de mi carrera pasaron a formar parte de mi vida, mis amigos, Yuni, Diuber, Yeni, Jan, Dany, Jova, Pedro, Lizzi, Ransés, Barriel, Kike, Benito, Osvaldito, Coto, Yarelis, Leonel, Francis, Bauza, Frank, Rosa, y Miguelito.*

*A todas aquellas personas que de una forma u otra han sabido ayudarme y brindarme su apoyo incondicional cuando lo he necesitado.*

*A todos muchas gracias.*

## RESUMEN

### RESUMEN

Los Sistemas de Información Geográfica son utilizados mundialmente en diferentes sectores de la sociedad, entre los que se destaca la prevención de delitos. Garantizar la seguridad ciudadana es una tarea de vital importancia para todos los países, las instituciones encargadas de resolver esta tarea para cumplir correctamente sus funciones, demandan constantemente información geográficamente referenciada para aumentar la calidad del proceso de toma de decisiones.

El Sistema de Prevención del Delito de Venezuela es una aplicación web, que tiene como objetivo el apoyo a la prevención del delito, mediante la gestión de la información y a las actividades fundamentales de las Coordinaciones Regionales adscritas a la Dirección General de Prevención del Delito del Ministerio del Poder Popular para Relaciones Interiores y Justicia de la República Bolivariana de Venezuela. Este sistema está estructurado por varios módulos que garantizan su correcto funcionamiento entre los que se encuentra el módulo Mapificación.

El módulo Mapificación tiene como finalidad ser una herramienta capaz de contribuir al proceso de toma de decisiones en el Sistema de Prevención del Delito de Venezuela. Permitirá representar reportes sobre mapas digitales con la información necesaria para monitorear, controlar y localizar eficientemente las acciones relacionadas con la prevención de delitos.

**PALABRAS CLAVE:** Sistemas de Información Geográfica, mapas, módulo Mapificación, prevención del delito.

# ÍNDICE

## ÍNDICE

|   |    |
|---|----|
| INTRODUCCIÓN.....   | 10 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....   | 15 |
| 1.Introducción.....   | 15 |
| 1.1 Sistemas de Información Geográfica en el mundo.....   | 15 |
| 1.1.1 El Mapa del Delito para la Ciudad Autónoma de Buenos Aires.....   | 15 |
| 1.1.2 Sistema Territorial de Emergencias y Gestión Policial de Chile (STEGPOL) .....  | 16 |
| 1.1.3 Mapa del delito en España.....  | 17 |
| 1.1.4 Mapa del delito en Caracas (Wikidelito Caracas) .....   | 17 |
| 1.1.5 Citizens Report.....  | 17 |
| 1.1.6 SIG del Departamento de Policía de Los Ángeles (LAPD).....  | 18 |
| 1.2 Sistemas de Información Geográfica en Cuba.....   | 18 |
| 1.2.1 Mapa del Delito de La Habana.....   | 19 |
| 1.2.2 Sistema Automatizado del Centro de Información y Mando de la Unidad Provincial de Patrullas de Ciudad de La Habana..... | 19 |
| 1.3 Sistemas de Información Geográfica en la UCI .....  | 19 |
| 1.3.1 SIG desarrollados sobre la plataforma GeneSIG del Centro GEYSED.....  | 20 |
| 1.3.2 SIG del Proyecto 171 del Centro ISEC.....   | 21 |
| 1.4 Conclusiones de los SIG analizados.....   | 21 |
| 1.5 Tecnologías y herramientas utilizadas en el desarrollo .....  | 22 |
| 1.5.1 Metodología de Desarrollo de Software.....  | 22 |
| 1.5.2 GeoServer 2.1.3. Servidor de Mapas.....   | 24 |
| 1.5.3 Open Layers 2.11.....   | 25 |
| 1.5.4 PostgreSQL 8.4. Sistema de Gestión de Bases de Datos (SGBD) .....   | 25 |
| 1.5.6 Servidor Apache Tomcat 7.0.....   | 27 |
| 1.5.7 Entorno de Desarrollo Integrado (IDE) .....   | 27 |
| 1.5.8 Plataforma de Desarrollo Java Platform Enterprise Edition (Java EE).....  | 28 |
| 1.5.9 Lenguajes de Programación .....   | 28 |
| 1.5.10 Frameworks.....  | 29 |
| 1.5.10.1 Spring Framework.....  | 29 |
| 1.5.10.2 Framework Hibernate .....  | 29 |
| 1.5.11 Visual Paradigm 8.0. Herramienta CASE.....   | 29 |
| 1.5.12 Lenguaje Unificado de Modelado (UML) .....   | 30 |
| 1.6 Conclusiones.....   | 30 |
| CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO MAPIFICACIÓN.....  | 31 |
| 2. Introducción.....  | 31 |
| 2.1 Descripción de la Solución Propuesta .....  | 31 |
| 2.2 Estructura del Mapa.....  | 32 |
| 2.3 Modelo Conceptual.....  | 33 |
| 2.3.1 Conceptos asociados al módulo Mapificación .....  | 33 |
| 2.4 Requisitos del Software .....   | 35 |
| 2.4.1 Requisitos Funcionales .....  | 35 |
| 2.4.2 Requisitos no Funcionales.....  | 36 |
| 2.5 Definición de los casos de uso.....   | 37 |

## ÍNDICE

|   |    |
|---|----|
| 2.5.1 Definición de los actores del Sistema.....                                    | 37 |
| 2.5.2 Definición de los casos de uso del sistema .....                              | 38 |
| 2.5.3 Diagrama de Casos de Uso .....  | 38 |
| 2.6 Descripción de los Casos de Uso (CU) .....                                      | 39 |
| 2.6.1 Descripción del CU Gestionar Capas .....                                      | 39 |
| 2.6.2 Descripción del CU Realizar Navegación.....                                   | 40 |
| 2.6.3 Descripción del CU Visualizar Reporte.....                                    | 40 |
| 2.7 Conclusiones.....   | 40 |
| CAPÍTULO 3: CONSTRUCCIÓN DEL MÓDULO MAPIFICACIÓN .....                              | 42 |
| 3. Introducción.....  | 42 |
| 3.1 Arquitectura base del Sistema de Prevención del Delito de Venezuela .....       | 42 |
| 3.1.2 Enfoque Horizontal.....   | 43 |
| 3.1.3 Enfoque Vertical.....   | 43 |
| 3.2 Diagrama de Clases del Análisis.....  | 44 |
| 3.3 Diagrama de colaboración .....  | 46 |
| 3.4 Patrones de Diseño .....  | 46 |
| 3.5 Diagrama de Clases del Diseño.....  | 48 |
| 3.6 Diagrama de Clases Entidades del Dominio .....                                  | 49 |
| 3.7 Modelo de Datos.....  | 50 |
| 3.8 Modelo de Implementación.....   | 51 |
| 3.8.1 Diagrama de Componentes.....  | 51 |
| 3.9 Modelo de Despliegue.....   | 53 |
| 3.9.1 Descripción de los Protocolos de Comunicación .....                           | 54 |
| 3.10 Conclusiones.....  | 54 |
| CAPÍTULO 4: PRUEBAS AL SISTEMA PROPUESTO .....                                      | 55 |
| 4. Introducción.....  | 55 |
| 4.1 Pruebas de Caja Blanca.....   | 55 |
| 4.2 Pruebas de Caja Negra.....  | 56 |
| 4.2.1 Pruebas de Caja Negra del CU Realizar Navegación .....                        | 56 |
| 4.2.2 Pruebas de Caja Negra del CU Gestionar Capas .....                            | 56 |
| 4.2.3 Pruebas de Caja Negra del CU Visualizar Reporte .....                         | 57 |
| 4.3 Resultados de las Pruebas aplicadas al módulo Mapificación .....                | 60 |
| 4.3.1 Resultados de las pruebas de caja negra aplicadas al módulo Mapificación..... | 60 |
| 4.4 Conclusiones.....   | 61 |
| CONCLUSIONES.....   | 62 |
| RECOMENDACIONES.....  | 63 |
| BIBLIOGRAFÍA.....   | 64 |
| GLOSARIO DE TÉRMINOS .....  | 68 |



## ÍNDICE

|   |    |
|---|----|
| ANEXOS .....                                      | 71 |
| Anexo 1 .Clases Entidades del Dominio .....       | 71 |
| Anexo 1.1 Clase Entidad Estado.....               | 71 |
| Anexo 1.2 Clase Entidad Municipio .....           | 71 |
| Anexo 1.3 Clase Entidad Denuncia .....            | 73 |
| Anexo 1.4 Clase Entidad Actividad .....           | 75 |
| Anexo 1.5 Clase Entidad ActividadPlanificada..... | 77 |
| Anexo 1.6 Clase Entidad ActividadRealizada .....  | 77 |

## INTRODUCCIÓN

### INTRODUCCIÓN

A través de la historia el hombre ha tenido la necesidad de orientarse en el terreno para poder definir recorridos, establecer distancias y especificar localizaciones. Las sociedades primitivas que vivían como cazadores y guerreros tenían que trasladarse de forma continua de un territorio hacia otro, por lo que en ocasiones era cuestión de vida o muerte conocer la trayectoria y dirección de los caminos y lugares a recorrer; por la gran necesidad de transmitirse unos a otros el conocimiento del terreno surgieron los primeros mapas.

El mapa más antiguo que existe apareció en Babilonia<sup>1</sup> hace miles de años antes de nuestra era (a.n.e), estaba grabado en una tableta de arcilla. La primera referencia a un mapa, semejante a los que conocemos hoy, proviene de una crónica china del año 227 a.n.e. en la que se relata cómo un individuo intentó asesinar al emperador con un puñal emponzoñado envuelto en un mapa. (1)

El arte de la cartografía se ha desarrollado y perfeccionado a través del tiempo. El amplio desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) en que se encuentra el mundo hoy, ha permitido la creación de diversas aplicaciones informáticas que optimizan el proceso de toma de decisiones en los diversos campos de la sociedad donde los mapas han jugado un papel importante ya que se puede representar sobre los mismos cualquier información útil al hombre.

Un Sistema de Información Geográfica (SIG) 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 geográfica. (2).

Debido a la importancia que han alcanzado los SIG en la actualidad se puede afirmar que son numerosas las áreas donde estos se aplican, entre los que se pueden citar: la planificación ambiental, geo-marketing, meteorología, administración de servicios públicos, ordenamiento territorial, manejo de rutas óptimas, sistemas de posicionamiento global y predicción, permitiendo en cada una de estas mejorar la calidad del proceso de la toma de decisiones.

Los SIG han ganado trascendencia en la prevención de delitos, ya que permiten combinar las bases de datos que contienen hechos delictivos con la información geográfica

---

<sup>1</sup> Babilonia: Ciudad importante de Mesopotamia, fundada en el año 1894 a.n.e.

## INTRODUCCIÓN

referenciada. Facilitan la realización de análisis espaciales y la creación de mapas de criminalidad para garantizar la correcta evaluación de los recursos y el diseño de programas, diagnósticos y actividades preventivas.

El Sistema de Prevención del Delito de Venezuela gestiona información sobre denuncias, actividades realizadas, atención de casos, planificación de actividades, gestión de diagnósticos y de la información asociada a la prevención del delito dividida en consejos comunales. Para la Dirección General de Prevención del Delito (DGPD) es necesario poder contar con un mapa interactivo, que le permita realizar análisis por áreas geográficas a partir de la información gestionada en el sistema, sobre el cual se necesitan aplicar estilos con el objetivo de diferenciar las distintas regiones del país de acuerdo a la cantidad de hechos registrados.

El módulo de mapas que hoy tiene el Sistema de Prevención del Delito de Venezuela está compuesto por una imagen estática que solo muestra los nombres de los estados del país. Este módulo no representa toda la información que se gestiona en el sistema, sino que solo visualiza la referente a los consejos comunales. Además, obliga a los usuarios a registrar la información sobre el mapa, ya que el sistema es incapaz de generarla a partir de los datos geo-referenciados almacenados en su base de datos. La forma en la que fue concebido impide cumplir con las necesidades de representación de la información que hoy tiene la DGPD. De igual manera esta solución no es escalable, impidiendo en el futuro mostrar nuevas informaciones en correspondencia con la evolución de los procesos y el sistema.

Lo antes expuesto permite declarar como **problema a resolver**: ¿Cómo mostrar información geo-referenciada asociada a la gestión de la prevención del delito en Venezuela?

Del problema planteado se deriva el siguiente **objeto de estudio**: Las herramientas para representar información sobre mapas digitales. Siendo el **campo de acción**: Los Sistemas de Información Geográfica. Se definió como **objetivo general**: Desarrollar el módulo Mapificación del Sistema de Prevención del Delito de Venezuela.

Los **objetivos específicos** que se derivan del objetivo general son:

- Caracterizar las tendencias actuales de la utilización de los sistemas informáticos en la prevención del delito.
- Diseñar una solución que contribuya a la gestión de la prevención del delito en Venezuela.

## INTRODUCCIÓN

- Implementar la propuesta de solución haciendo uso de estándares y buenas prácticas.
- Aplicar pruebas para validar la solución.

Para lograr el cumplimiento de los objetivos expuestos anteriormente se desarrollarán estas tareas en la presente investigación:

- Caracterización de los Sistemas de Información Geográficas para evaluar la forma de representación de información a través de mapas digitales.
- Caracterización de los principales servidores de mapas de código abierto utilizados en el mundo para elegir el más adecuado a las condiciones actuales del Sistema de Prevención del Delito de Venezuela.
- Selección de la cartografía para representar las capas base de la solución.
- Refinamiento de los requisitos existentes para el módulo Mapificación para cumplir con las expectativas del cliente.
- Definición de los nuevos requisitos para añadir valor agregado al sistema.
- Realización del modelo de casos de uso del sistema para describir el funcionamiento del sistema de acuerdo a los requisitos.
- Diseño del módulo de Mapificación para definir la estructura y relaciones de los componentes del sistema.
- Implementación del módulo de Mapificación para dar solución a los requisitos.
- Diseño de casos de prueba para ensayar escenarios de la aplicación.
- Realización de Pruebas al módulo de Mapificación para validar que la aplicación satisface las necesidades del cliente.

### **Posibles resultados:**

- Modelo de diseño del módulo de Mapificación del Sistema de Prevención del Delito de Venezuela.
- Modelo de Implementación del módulo de Mapificación del Sistema de Prevención del Delito de Venezuela.
- Módulo de Mapificación del Sistema de Prevención del Delito de Venezuela.

## INTRODUCCIÓN

Durante la investigación se emplearon los siguientes **métodos científicos**:

### **Métodos teóricos:**

- **Analítico-sintético:** Utilizado para ejecutar un análisis de la bibliografía correspondiente a la investigación y extraer los elementos significativos para el diseño eficiente del SIG del Sistema de Prevención del Delito de Venezuela.
- **Modelación:** Se pone en práctica para la realización de los diagramas correspondientes a la fase de análisis y diseño, proporcionando una mejor perspectiva del SIG que se quiere desarrollar.

### **Métodos empíricos:**

- **Entrevista:** Se utilizó para realizar entrevistas al tutor y a trabajadores del grupo de desarrollo del Sistema de Prevención del Delito de Venezuela para definir los elementos y requisitos que necesita el módulo Mapificación.

El presente documento refleja cuatro partes fundamentales, resumen, introducción, desarrollo y conclusiones. El desarrollo está constituido por cuatro capítulos los cuales están organizados de la siguiente manera:

### **Capítulo 1: Fundamentación Teórica**

Se analizan las tendencias y tecnologías actuales a nivel mundial, concernientes a los Sistemas de Información Geográfica con el fin de definir el servidor de mapas que se va a utilizar en el desarrollo del módulo. Se fundamenta el uso de las herramientas, tecnologías, lenguaje de programación y metodología de desarrollo de software que se utiliza, lo que constituye el basamento teórico de la investigación.

### **Capítulo 2: Características del módulo Mapificación**

Se elabora un modelo de dominio donde se analizan las entidades y conceptos presentes en el contexto donde se desarrolla el módulo Mapificación, se definen las relaciones existentes entre cada uno de estos. Se especifican los requisitos funcionales y no funcionales con las que debe cumplir el sistema, se muestra el diagrama de casos de uso del sistema, se describen los actores y se describen los casos de uso del sistema.

### **Capítulo 3: Construcción del módulo Mapificación**

Se abordan todos los aspectos relacionados con el diseño e implementación del módulo Mapificación. Se muestran los diagramas de: clases del análisis, colaboración, clases del diseño, y el de clases entidades del dominio. Se explica la arquitectura del Sistema de Prevención del Delito de Venezuela y los patrones de diseño que se utilizaron en la construcción del módulo Mapificación.

## INTRODUCCIÓN

### **Capítulo 4: Pruebas al Sistema Propuesto**

Se exponen todos los aspectos relacionados con las pruebas de caja negra y pruebas de caja blanca efectuadas al módulo Mapificación con el objetivo de identificar errores.

## CAPÍTULO 1

### CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

#### 1. Introducción

En el presente capítulo se analizan las tendencias y tecnologías novedosas existentes en el mundo, relacionadas con los Sistemas de Información Geográfica persiguiendo el objetivo de definir el servidor de mapas que se va a utilizar en el desarrollo del módulo. Se fundamentan las tecnologías, herramientas, lenguajes de programación y metodología de desarrollo de software a utilizar en la implementación del sistema, lo que constituye la base teórica de la investigación.

#### 1.1 Sistemas de Información Geográfica en el mundo

Los SIG son herramientas que han logrado desarrollarse con el transcurso del tiempo, ya que son de gran beneficio para investigar y recopilar información con el fin de solucionar problemas existentes en la sociedad.

El primer SIG verdaderamente operacional apareció en Canadá, Ottawa, en 1962 y fue desarrollado por Roger Tomlinson del Departamento canadiense de Bosques y Desarrollo Rural, en un esfuerzo de usar superposiciones del mapa para el análisis de las diversas áreas de Canadá. Esta primera versión se llamó CGIS. (3)

Son numerosas las áreas de la sociedad donde se emplean los SIG, por lo que la investigación se centrará en analizar los creados en la Universidad de las Ciencias Informáticas (UCI) y solo los destinados a la prevención del delito desarrollados en Cuba y en el mundo, como son: el Mapa del Delito de la ciudad autónoma de Buenos Aires, Sistema Territorial de Emergencias y Gestión Policial de Chile, Mapa del Delito en España, Mapa del Delito en Caracas, Citizens Report, SIG del Departamento de Policía de Los Ángeles (LAPD), Mapa del Delito de La Habana y el Sistema Automatizado del Centro de Información y Mando de la Unidad Provincial de Patrullas de Ciudad de La Habana, los SIG desarrollados en el Centro de Geoinformática y Señales Digitales (GEYSED), y el SIG del Proyecto 171 perteneciente al Centro de Informatización de la Seguridad Ciudadana (ISEC). Algunos de estos SIG se describirán a continuación.

##### 1.1.1 El Mapa del Delito para la Ciudad Autónoma de Buenos Aires

Surge en Argentina a partir de un convenio entre el Centro de Información Metropolitana de la Facultad de Arquitectura, Diseño y Urbanismo y el Ministerio Público Fiscal de la nación. Cuenta con una base de datos de los hechos delictivos de autoría desconocida, registrados

## CAPÍTULO 1

en la Capital Federal. Esta base contiene información referida al hecho, como ser: tipo de delito, fecha, lugar y cantidad de víctimas.

La existencia de este SIG no es muy conocida y tiene como limitante que solo abarca hechos ocurridos en Capital Federal con autoría desconocida.(4)

Se tuvo como meta identificar las zonas calientes<sup>2</sup> con el fin de visualizar la situación actual de la distribución del delito, detectar y analizar tendencias de concentración y evolución, a partir de contar con datos oficiales, para orientar políticas de seguridad. Dándole seguimiento a las zonas críticas para evitar situaciones de oportunidad a través de una distribución de recursos que mejore la eficacia del quehacer policial. (5)

Este SIG fue desarrollado con las siguientes tecnologías:

- Lenguaje de programación: Visual Basic.
- Servidor de Mapas: ArcView3.2.

Este SIG sirvió como ejemplo para el desarrollo del módulo Mapificación en la forma de representar las áreas que posean la mayor cantidad de ocurrencias de un reporte.

Se desarrolló utilizando tecnologías privativas las cuales no pueden ser utilizadas para la implementación del módulo Mapificación debido a las políticas existentes tanto en Cuba como en Venezuela de promover la creación de software libres con el objetivo de garantizar la soberanía tecnológica de la nación.

### **1.1.2 Sistema Territorial de Emergencias y Gestión Policial de Chile (STEGPOL)**

Es un sistema implementado sobre plataforma web, el principal objetivo de este proyecto fue crear un SIG con tecnologías novedosas, que funcionara sobre una plataforma nacional común de información aplicada al Sistema de Emergencias Nacionales y al Sistema Territorial de Gestión Policial que actualmente operan en Chile.

La idea principal del Proyecto STEGPOL es integrar a las diferentes entidades policiales como Carabineros, Investigaciones, Ministerio del Interior y Municipios, entre otros, en una plataforma nacional común de información que permita el intercambio de datos y que sirva de apoyo a la gestión operacional regional o comunal, donde dichas instituciones estén interconectadas entre sí, en especial con diferentes divisiones o departamentos de Carabineros, tales como Jefaturas de Zona, Prefecturas, Comisarías, Sub-Comisarías,

---

<sup>2</sup> zonas calientes: Representa un área localizada que contiene una gran porción del volumen total de incidentes delictivos de toda esa región. (5)



## CAPÍTULO 1

Tenencias y por último los Retenes, más el apoyo directo y coordinado entre Carabineros y el área de Seguridad de los Municipios. (6)

No existe información pública de las tecnologías con las que se desarrolló este sistema. Es un SIG que sirve como ejemplo de integración varias entidades policiales de los distintos lugares del país en una única plataforma de información.

### 1.1.3 Mapa del delito en España

Es un mapa del delito desarrollado sobre plataforma web por la empresa Mapalia Networks, con el objetivo de crear un mapa de la delincuencia en España que ayude a detectar cuáles son las zonas más peligrosas y conflictivas y cuáles las más tranquilas. Para ello, la información que sirve de base a esta nueva aplicación procede de fuentes policiales y medios de comunicación.

La iniciativa lanzada por Mapalia Networks hace especial hincapié en la funcionalidad de este servicio para que los ciudadanos conozcan de primera mano cuáles son los problemas de seguridad que les rodean. (7)

No existe información pública de las tecnologías y herramientas que se utilizaron para desarrollar este sistema. Es un SIG que sirve como ejemplo en la forma que representa en el mapa las zonas de mayor y menor cantidad de incidencias de delitos.

### 1.1.4 Mapa del delito en Caracas (Wikidelito Caracas)

Desarrollado por la Compañía Ushahidi para que las personas puedan denunciar delitos que le hayan ocurrido como: robos, secuestros, homicidios, agresiones, abuso de autoridad, tráfico de drogas, entre otras; además brinda la opción de establecer la fecha y ubicar en el mapa el lugar aproximado de la infracción. Es un SIG implementado sobre plataforma web, no existe información pública de las tecnologías utilizadas.

Este SIG no muestra ninguna información de las acciones, programas y actividades que se llevan a cabo en este país para prevenir los delitos; solo ofrece la posibilidad de denunciar delitos ocurridos, además no existe la forma de probar la veracidad de las denuncias realizadas por los usuarios ni la identidad de los mismos ya los reportes de denuncias pueden realizarse de forma anónima.

### 1.1.5 Citizens Report

Es un sitio web creado por la Compañía Shaka Zoom para ofrecer un portal abierto a los ciudadanos de Reino Unido, con el fin de que puedan crear su propia comunidad de

## CAPÍTULO 1

inteligencia y alertar a otros sobre la delincuencia y problemas perjudiciales en sus vecindarios. Brinda la posibilidad hacer informes criminales de forma instantánea y anónima en los cuales se permite describir y ubicar los delitos. Este portal tiene como característica no comunicar a la policía u otros servicios de justicia penal de los informes contenidos en sus bases de datos a menos que se lo exijan. (8)

No existe información pública de las tecnologías y herramientas utilizadas para el desarrollo de este sistema. Este sistema tiene como desventaja que representa varios reportes utilizando la misma simbología en el mapa, los cual podría causar confusión a los usuarios que utilicen el mismo.

### 1.1.6 SIG del Departamento de Policía de Los Ángeles (LAPD)

Es un SIG implementado sobre plataforma web, realizado por el Grupo Omega para ser utilizado por el Departamento de Policía de Los Ángeles (LAPD) con el propósito de combatir y reducir la delincuencia. Este SIG es accesible para la población, permite realizar y visualizar reportes sobre violaciones, asaltos, asesinatos y delitos contra la propiedad. Con su surgimiento puso fin a la pausa del mapeo de la delincuencia que existía debido a los errores de la aplicación de mapas que usaba este departamento, la cual fue confeccionada por Lightray Productions.

Este SIG fue desarrollado con tecnologías privativas, utilizando el conjunto de productos de software ArcGIS el cual está producido y comercializado por la compañía Environmental Systems Research Institute (ESRI) y un API de JavaScript que conecta con el servidor al servidor de mapas. (9)

En la interfaz de usuario (IU) de este SIG sirve como ejemplo para el módulo que se desea implementar, ya que se muestran los detalles de todos los reportes que se visualizan en el mapa, permitiéndole a los usuarios una fácil interacción con el sistema.

### 1.2 Sistemas de Información Geográfica en Cuba

En Cuba la seguridad de los ciudadanos es de vital importancia, ya que se diseñan medidas y acciones con el objetivo de evitar conductas o malos comportamientos que puedan convertir a uno o varios miembros de la población en víctimas de algún hecho delictivo. Entre las medidas que se han tomado se encuentra la creación de varios SIG con la intención de mejorar el proceso de toma de decisiones en entidades responsabilizadas de la prevención del delito en la nación.

## CAPÍTULO 1

### 1.2.1 Mapa del Delito de La Habana

Es una aplicación de escritorio perteneciente al Ministerio del Interior (MININT), tiene como objetivo mostrar la información geográfica referente a las Fuerzas en Servicio y al Estado de Criminalidad de la ciudad, dentro de sus funcionalidades más importantes se encuentra la representación de un mapa compuesto por diferentes capas (la capa de las calles de Ciudad de La Habana, la capa de los municipios, la capa de las áreas de patrullaje), filtración de órdenes de radio a través de criterios de búsqueda especializada y la representación de resultados en el mapa.

Este SIG fue desarrollado utilizando las siguientes tecnologías: Visual Basic 6.0, MapInfo 6.1, MapBasic y Oracle 9i; estas tecnologías son privativas y en el caso de Oracle 9i su licencia posee un alto costo. (10)

Como las tecnologías que se emplearon para su desarrollo son privativas no pueden ser utilizadas para la implementación del módulo Mapificación.

Este sistema es un ejemplo de cómo se puede mostrar sobre un mapa digital las acciones y actividades que se realizan en un país para prevenir delitos.

### 1.2.2 Sistema Automatizado del Centro de Información y Mando de la Unidad Provincial de Patrullas de Ciudad de La Habana

Este SIG al igual que el Mapa del Delito de La Habana, es una aplicación de escritorio perteneciente al MININT, desarrollado con las mismas tecnologías y poseen funcionalidades similares, lo diferencia la representación en el mapa de la ubicación de objetivos de interés para la Administración del Estado, misiones especiales y los planes operativos de esta unidad. (10)

### 1.3 Sistemas de Información Geográfica en la UCI

La Universidad de las Ciencias Informáticas a lo largo de los años ha alcanzado gran experiencia en el desarrollo de proyectos de diversas áreas de la informática.

El Centro GEYSED de la Facultad 6 es el encargado de desarrollar productos, servicios y soluciones informáticas en área de la Geoinformática, este centro ha creado numerosas aplicaciones utilizando la plataforma soberana (GeneSIG) la cual persigue fortalecer la experiencia en la realización de SIG y abrir un espacio sólido en el mercado de aplicaciones de esta rama.

## CAPÍTULO 1

A continuación se muestra una descripción de los SIG desarrollados por el Centro GEYSED utilizando la plataforma GeneSIG, y el SIG del Proyecto 171 perteneciente al Centro ISEC de la Facultad 2.

### 1.3.1 SIG desarrollados sobre la plataforma GeneSIG del Centro GEYSED

La plataforma GeneSIG sobre la cual están implementados los SIG del Centro GEYSED es una plataforma implementada con herramientas y tecnologías libres. Cumple técnicamente con las especificaciones OpenGIS que establece el Open Geospatial Consortium (OGC) que garantizan la interoperabilidad global entre los SIG y con la política de migración a software libre que se impulsa en Cuba con el objetivo de lograr soberanía tecnológica. (11)

A continuación se muestran las herramientas y tecnologías necesarias para usar la plataforma GeneSIG:

- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.
- MapServer 5.2.2 o superior con todas sus dependencias, con la extensión PHP mapscript.
- Lenguaje de Programación PHP.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsqllite incluida.
- PgRouting como componente de PostGIS para el análisis de rutas (en caso de requerir la funcionalidad).

Productos desarrollados sobre la plataforma GeneSIG:

- SIGUCI.
- SIGRUTAS.
- SIGSALUD.

Estos SIG se caracterizan por realizar las funcionalidades básicas (acercar, alejar, centrar, alejar a la máxima extensión y mover el mapa permitiendo la visualización de objetos entre las distintas escalas disponibles) y son software que se centran en cumplir las necesidades de los clientes.

La plataforma GeneSIG ha demostrado que se pueden desarrollar SIG con la calidad requerida utilizando tecnologías libres de costo.

## CAPÍTULO 1

El estudio de la plataforma GeneSIG aportó conocimientos sobre herramientas y tecnologías libres que se utilizan la creación de SIG (OpenLayers, PostgreSQL, PostGIS y MapServer) así como la formar en que representan la información sobre mapas digitales.

El servidor de mapas que utilizan es MapServer el cual no posee una interfaz de administración, le extensibilidad del mismo está orientada a los desarrolladores que utilicen el lenguaje PHP, soporta los servicios WMS, WFS y WCS. (36)

Esta plataforma no se puede utilizar en el desarrollo del módulo Mapificación porque afectaría el ambiente de despliegue del sistema, ya que se necesita para un servidor Apache versión 2.0 para despliegue de la misma.

### 1.3.2 SIG del Proyecto 171 del Centro ISEC

El Centro de Informatización de la Seguridad Ciudadana (ISEC) de la Facultad 2 desarrolló el SIG del proyecto 171 el cual es una aplicación de escritorio, implementada en Lenguaje Java y como servidor de Mapas se usó Map Xtreme Java Edition, es un software a la medida que se enfoca en cumplir las necesidades del cliente.

Este SIG usa un servidor de mapas que es privativo por lo cual no puede ser utilizado en el desarrollo del módulo Mapificación. Sirvió como ejemplo de SIG desarrollado con el lenguaje de programación Java, además estimuló el estudio de servidores de mapas (GeoServer) que fueran libres y compatibles con este lenguaje ya que es el definido para desarrollar módulo.

### 1.4 Conclusiones de los SIG analizados

Con el análisis realizado a los SIG se detectó que muchos de estos están creados con tecnologías privativas, las cuales no pueden utilizarse en el desarrollo del módulo Mapificación debido a los costos por conceptos de licencias y a las políticas existentes en Cuba y en Venezuela de promover la creación de software utilizando tecnologías de código abierto con el objetivo de alcanzar la soberanía tecnológica del país.

Se decidió que el módulo Mapificación se implementara utilizando tecnologías de código abierto, ya que es factible la construcción de sistemas de calidad utilizando estas tecnologías, debido a la existencia de una amplia gama y variedad libres de costo, copia, redistribución y modificación.

El Sistema de Prevención del Delito de Venezuela necesita un SIG que constituya un módulo, que use su misma arquitectura, por lo tanto no se pueden usar ninguno de los sistemas desarrollados en la UCI, de estos se tomaron las ventajas que ofrecían (la forma

## CAPÍTULO 1

de representación de la información sobre el mapa y el uso de herramientas y tecnologías libres de costo para desarrollar SIG) que sirvieron como base de conocimientos para la implementación del módulo Mapificación.

Se valoraron las deficiencias de representación de la información sobre los mapas digitales que tenían algunos de los SIG analizados, con el objetivo de no cometer estos errores en la implementación del módulo Mapificación.

### 1.5 Tecnologías y herramientas utilizadas en el desarrollo

La apropiada selección de las herramientas y tecnologías a utilizar en la creación de un software posee estrecha relación con el tiempo de desarrollo y la calidad final del producto.

El módulo Mapificación su acoge a la arquitectura del Sistema de Prevención del Delito de Venezuela por lo que utiliza las herramientas, tecnologías y flujos de procesos definidos para desarrollar este sistema, excepto el servidor de mapas y la biblioteca JavaScript OpenLayers, que fueron introducidas para el desarrollo de este módulo.

#### 1.5.1 Metodología de Desarrollo de Software

**Rational Unified Process (RUP):** Metodología de desarrollo de software que provee una orientación disciplinada para asignar tareas y responsabilidades dentro de una organización que produce software. Su meta es asegurar la realización del software con la calidad requerida para resolver las necesidades de los usuarios finales y clientes dentro de un presupuesto y tiempo establecidos. (12)

RUP es un proceso que se caracteriza por:

- **Dirigido por casos de uso.** Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario y se usan para determinar el alcance de cada iteración y el contenido de trabajo de cada persona del equipo de desarrollo. (13)
- **Centrado en la arquitectura.** La arquitectura permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto. (13)
- **Iterativo e incremental.** Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición, y cada una de ellas se divide en iteraciones. En cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas propuestas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, Gestión de cambio y configuraciones, Gestión de Proyecto y Entorno. Cada iteración añade funcionalidades al producto de software o mejora las existentes. (13)

## CAPÍTULO 1

Los flujos de trabajos y el peso que tienen en cada una de las fases que proporciona RUP para desarrollar de forma eficaz un producto de software de calidad se exponen en la Figura 1.

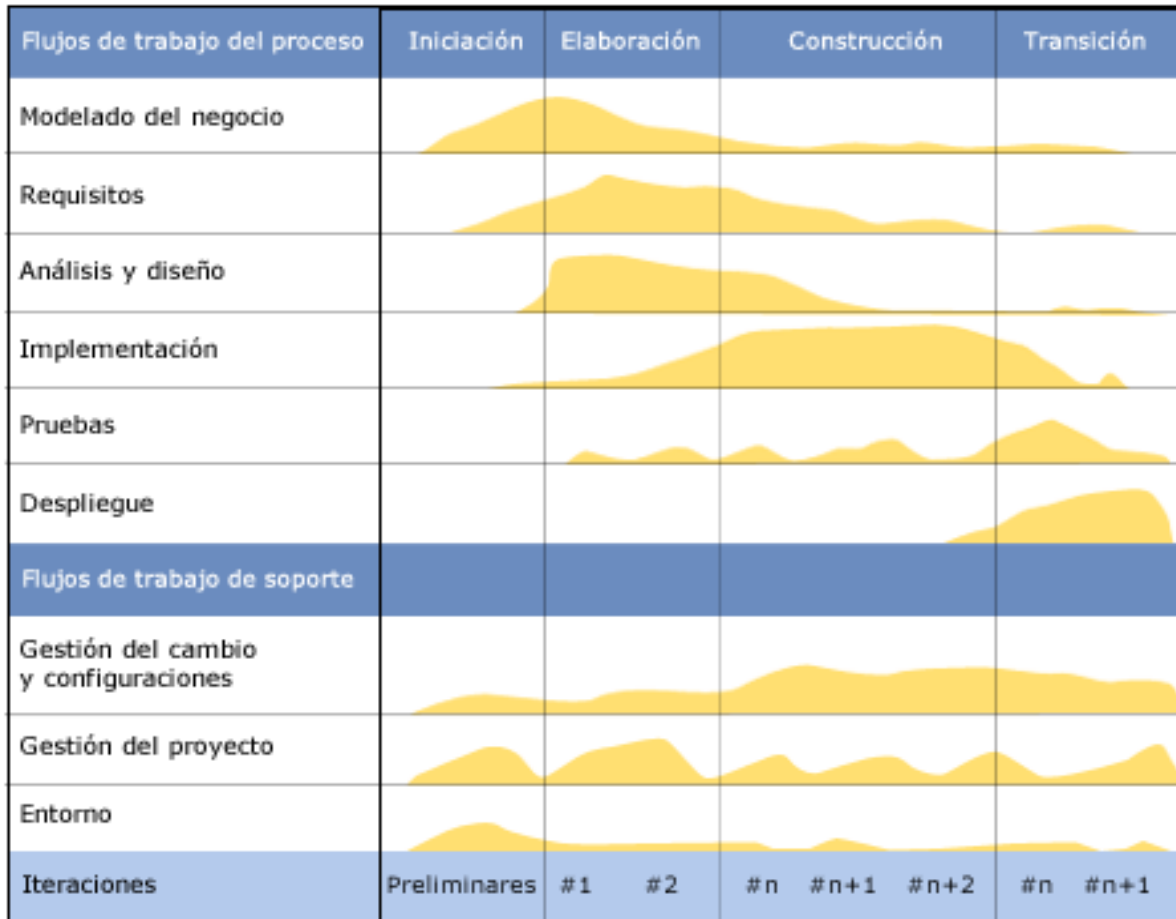


Figura 1. Flujos de Trabajos y Fases de la Metodología RUP.

En la fase de Inicio las iteraciones hacen mayor énfasis en las actividades de modelado del negocio y los requisitos; en la de elaboración las iteraciones se orientan a desarrollar la línea base de la arquitectura, comprenden más los flujos de trabajo de requisitos, modelado de negocio, análisis y diseño y una parte de implementación orientado a la línea base de la arquitectura.

En la de construcción, se comienza la construcción del producto de software por medio de una serie de iteraciones, para cada iteración se eligen los casos de uso, se refina su análisis y diseño y se procede a la implementación y pruebas; en la de transición se pretende garantizar que se tiene un producto dispuesto para su entrega al cliente y a los usuarios finales. Como se puede ver en cada fase intervienen todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

## CAPÍTULO 1

El Sistema de Prevención del Delito de Venezuela se desarrolló siguiendo esta metodología, por lo tanto en el desarrollo del módulo Mapificación se generan y consumen artefactos basados en ella.

### 1.5.2 GeoServer 2.1.3. Servidor de Mapas

El servidor de mapas escogido para desarrollar el módulo Mapificación es GeoServer, ya que es un software de código abierto, desarrollado en Java, permite a los usuarios compartir y editar datos geoespaciales y está diseñado para publicar datos de cualquier fuente significativa de datos espaciales usando estándares abiertos.

Al ser un proyecto impulsado por la comunidad, GeoServer es desarrollado, probado y apoyado por un grupo diverso de individuos y organizaciones de todo el mundo; es la implementación de referencia del Open Geospatial Consortium (OGC), y de los estándares Web Feature Service (WFS) y Web Coverage Service (WCS). Posee un alto rendimiento certificado de los Web Map Service (WMS). GeoServer constituye un componente esencial de la Web Geo-espacial. (14)

Es el servidor que va a alojar las imágenes vectoriales de los estados y municipios de Venezuela que van a ser utilizadas por el módulo Mapificación.

Algunas características que posee:

- Totalmente compatible con las especificaciones WMS (1.1.1 y 1.3), WFS (1,0 y 1,1, transacciones y bloqueo) y WCS (1,0 y 1,1) que se utilizan en el desarrollo del módulo Mapificación para consumir los servicios que ofrece GeoServer.
- Sirve como implementación de referencia para WCS 1,1 y 1,0.
- Formato de entrada de imagen: PostGIS, Oracle Spatial, ArcSDE, DB2, MySQL, Shapefile, GeoTIFF, GTOPO30, ECW, MrSID, JPEG2000. En el desarrollo del módulo Mapificación se utiliza el formato de entrada de imagen Shapefile, porque en este formato se encuentra la cartografía geo-referenciada correspondiente a los estados y municipios de Venezuela.
- Formato de salida de imagen: JPEG, GIF, PNG, PDF, SVG, Shapefile y KML. En el desarrollo del módulo Mapificación se utilizan los formatos de salida de imagen JPEG, PNG y GML.
- Soporte nativo para Java GeoTIFF, GTOPO30 e Image Mosaics.
- Java (J2EE) basado en servlets, se puede ejecutar en cualquier contenedor de servlets, es una ventaja porque se puede desplegar en el mismo contenedor web que el Sistema de Prevención del Delito de Venezuela.



## CAPÍTULO 1

- Fácil integración con la biblioteca OpenLayers.
- Soporte completo de SLD, como definiciones del usuario (POST y GET), y como uso de configuración de estilos. Esta característica se usa para solicitar y aplicar estilos a las capas que contiene el servidor.

### 1.5.3 Open Layers 2.11

OpenLayers es una biblioteca JavaScript de código abierto, fue seleccionada para desarrollar el módulo Mapificación porque permite del lado del cliente hacer mapas interactivos y visibles en casi todos los navegadores web, fue creada originalmente por MetaCarta, como respuesta, en parte, a Google Maps. Es muy conocida por muchos desarrolladores y posee una gigantesca comunidad de ayuda. (15)

La capa de presentación del módulo Mapificación se basa en el uso de esta biblioteca ya que facilita una IU simplificada que solicita los servicios WMS y WFS al servidor de mapas GeoServer de forma transparente para los usuarios y desarrolladores.

Se caracteriza por su simplicidad de uso ya que con poco desarrollo se agregan los controles e invocar las funciones necesarias para manipular el trabajo con cada una de las capas que conforman el mapa; él soporta tiles y caché, y además permite el acceso a muchos servidores de mapas (GeoServer, MapServer, Google Maps, entre otros). Posee una potente integración con GeoServer para visualizar las capas contenidas en el servidor. (16)

### 1.5.4 PostgreSQL 8.4. Sistema de Gestión de Bases de Datos (SGBD)

Sistema gestor de base de datos libre, publicado bajo la licencia BSD, fue seleccionado porque soporta datos espaciales mediante la extensión PostGIS, conjuntamente con la gestión de objetos geográficos. Este aspecto es principal dadas las particularidades del SIG a desarrollar. (17)

PostgreSQL está definido como el sistema gestor de base de datos a utilizar para la elaboración y despliegue del Sistema Informático de Gestión de Información de las Coordinaciones Regionales de Prevención del Delito ya que posee características como:

- Robustez, cumple con los estándares SQL.
- Optimizador de consultas.
- Multiplataforma (Linux, Windows, Mac OS X, Solaris, y otros más).
- Ofrece una disponibilidad, fiabilidad y seguridad.

## CAPÍTULO 1

- Permite Copias de seguridad en caliente (Online/hot backups).
- Es una base de datos 100% ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad).

**PostGIS:** Es una extensión de PostgreSQL para dar soporte a datos espaciales (Puntos, Líneas y Polígonos). (18)

Se utiliza para invocar las funciones que ofrece para trabajar con la cartografía correspondiente a los estados y municipios de Venezuela.

Principales características:

- Hace posible usar todos los objetos que aparecen en las especificaciones OpenGIS (puntos, líneas, polígonos, multipuntos y colecciones geométricas).
- Posee soporte de datos ráster<sup>3</sup> y vectoriales<sup>4</sup>.
- Tiene compatibilidad con SQL/MM Spatial<sup>5</sup>.
- Cuenta con una administración simplificada (lo que permite una mayor facilidad en la creación de tablas y vistas).

### 1.5.5 Java Persistence API (JPA)

Es una especificación de la empresa Sun Microsystems para la persistencia de objetos Java a cualquier base de datos relacional. Proporciona un modelo de persistencia que permite mapear bases de datos relacionales en Java. Puede ser utilizado directamente en aplicaciones tanto web como clientes.

El mapeo objeto/relacional, es decir, la relación entre entidades Java y tablas de la base de datos, se efectúan mediante anotaciones en las propias clases de entidad, por lo que no se necesitan ficheros descriptores XML. Además pueden definirse transacciones como anotaciones JPA.

Java Persistence API consta de tres áreas:

- El Java Persistence API.
- El lenguaje de query.

---

<sup>3</sup> Datos ráster: son una abstracción de la realidad, representan ésta como una rejilla de celdas o píxeles, en la que la posición de cada elemento es implícita según el orden que ocupa en dicha rejilla.

<sup>4</sup> Datos vectoriales: conjunto de líneas o arcos, definidos por sus puntos de inicio y fin, y puntos donde se cruzan varios arcos, los nodos. La localización de los nodos y la estructura topológica se almacena de forma explícita.

<sup>5</sup> SQL/MM Spatial: estándar para administrar datos espaciales en sistemas de bases de datos relacionales.

## CAPÍTULO 1

- El mapeo de los metadatos objeto/relacional. (19)

Se usa en el desarrollo del módulo Mapificación como un estándar de acceso a datos.

### 1.5.6 Servidor Apache Tomcat 7.0

Apache Tomcat es el contenedor web determinado para el desarrollo y despliegue del Sistema de Prevención del Delito de Venezuela. Es una implementación de software de código abierto de Java Servlets y tecnologías JavaServer Pages (JSP), es multiplataforma, funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jakarta perteneciente a la Apache Software Foundation bajo la licencia Apache 2.0. (20)

Posee las siguientes particulares:

- Autenticación de acceso básico.
- Negociación de credenciales.
- Protocolo de transferencia de hipertexto seguro (HTTPS).
- Alojamiento compartido.
- CGI o interfaz de entrada común.
- Servlets de Java.
- Server-Side Includes (SSI).
- Consola de administrador.

En este contenedor web se va a permitir alojar a dos herramientas fundamentales del módulo Mapificación: el servidor de mapas GeoServer y la biblioteca JavaScript OpenLayers.

### 1.5.7 Entorno de Desarrollo Integrado (IDE)

**Eclipse:** Es el entorno de desarrollo definido para desarrollar el Sistema de Prevención del Delito de Venezuela. Permite la refactorización y la actualización/instalación automática de código. Es completamente neutral a la plataforma y al lenguaje. Soporta diversos lenguajes como: Java, C/C++, Python, Eiffel, PHP, Ruby, y C#. (21)

Posee una arquitectura de plug-ins y un API que proporciona el plug-in Development Environment para ampliarlo. Añade soporte para un nuevo tipo de editor, una vista, o un lenguaje de programación, es fácil de usar, con el API y los bloques de construcción que proporciona Eclipse. (21)

## CAPÍTULO 1

### 1.5.8 Plataforma de Desarrollo Java Platform Enterprise Edition (Java EE)

Java EE posee con una arquitectura de varios niveles distribuida basándose ampliamente en componentes de software modulares, ejecutándose sobre un servidor de aplicaciones. Es considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes al mismo.

Java EE configura algunas especificaciones únicas para componentes, estas incluyen: Enterprise JavaBeans, Servlets, JavaServer Pages y varias tecnologías de servicios web. Esto permite al desarrollador crear una aplicación de empresa portable entre plataformas, y a la vez que sea integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel. (19)

Debido a que las Coordinaciones Regionales que van a usar el Sistema de Prevención del Delito de Venezuela están ubicadas en los diferentes estados de Venezuela, se hace necesario un sistema orientado a la web, por este motivo el colectivo de autores decide utilizar la Plataforma Java EE.

### 1.5.9 Lenguajes de Programación

**Java:** Lenguaje orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas, es compilado e interpretado, diseñado para crear software altamente fiable.

Provee numerosas comprobaciones en compilación y en tiempo de ejecución, está creado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. (22)

**JavaScript:** Lenguaje que no requiere de compilación ya que trabaja del lado del cliente. Los navegadores son los encargados de interpretar el código, por lo que no se requiere tener instalado ningún framework. Principalmente es usado para la incorporación de comportamiento dinámico a documentos HTML (describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas). (23)

## CAPÍTULO 1

La capa de presentación del módulo Mapificación se basa en gran medida de la utilización de código JavaScript y de la biblioteca OpenLayers la cual está implementada en este lenguaje.

### 1.5.10 Frameworks

Un framework (marco de trabajo) es una mini arquitectura reusable que provee una estructura y comportamiento genéricos para una familia de abstracciones de software. (24) Es un agregado de componentes orientado a dar solución a un tipo de problemas, sirve como referencia para desarrollar aplicaciones informáticas, brinda un conjunto amplio de funcionalidades lo que implica ahorro de tiempo en la implementación de sistemas informáticos.

#### 1.5.10.1 Spring Framework

Spring es un framework de código abierto que posibilita conformar aplicaciones robustas a partir de componentes simples, proporciona soporte para la programación orientada a aspectos, está dividido en módulos bien definidos pero no obliga a hacer uso de todos ellos, ya que facilita la elección de los módulos que se necesiten en el caso específico y buscar otras opciones cuando Spring no satisfaga los requisitos. (25)

El Sistema Prevención del Delito está implementado sobre este framework y se utilizan los módulos Spring Security y Spring MVC.

#### 1.5.10.2 Framework Hibernate

Es un framework distribuido bajo los requisitos de la licencia GNU LGPL, es una herramienta que ha alcanzado popularidad ya que brinda soporte a la capa de acceso a datos en el desarrollo de aplicaciones empresariales. Proporciona las siguientes características: mapeo objeto-relacional básico, caché, caché distribuida, carga perezosa y ávida, entre otras. (26)

Se usa en el Sistema de Prevención del Delito de Venezuela como framework de acceso a datos.

### 1.5.11 Visual Paradigm 8.0. Herramienta CASE

Es una herramienta CASE profesional, fácil de utilizar y que soporta el ciclo de vida completo del desarrollo de software. Posee una alternativa libre y gratuita. Presenta muchas facilidades que permiten crear diagramas de:

- Casos de Uso.

## CAPÍTULO 1

- Procesos del Negocio.
- Clases.
- Estado.
- Despliegue.
- Comunicación.
- Secuencia.
- Componentes.
- Objetos.
- Entidad Relación.
- Interacción.
- ORM.
- Visión general.

Se usa en el desarrollo del módulo Mapificación para la creación de los diagramas utilizando el Lenguaje Unificado de Modelado.

### **1.5.12 Lenguaje Unificado de Modelado (UML)**

UML es un lenguaje estándar para especificar, visualizar, construir y documentar los artefactos de sistemas informáticos, así como para el modelado de negocios y otros sistemas. (27)

UML ofrece diversos elementos de esquematización que representan las diferentes partes de un sistema, de los cuales se utilizan en el desarrollo del módulo Mapificación los diagramas de casos de uso, de clases del diseño, de despliegue, etc.

### **1.6 Conclusiones**

Con el análisis realizado a varios SIG desarrollados en el mundo, en Cuba y en la UCI, se logró caracterizarlos teniendo en cuenta las herramientas, tecnologías y las formas de representación de la información geo-referenciada empleadas para su desarrollo, permitiendo así seleccionar las que más se adecuan a las condiciones actuales del Sistema de Prevención del Delito de Venezuela.

Se mencionan las tecnologías a utilizar para el desarrollo del sistema a implementar, donde se propone, teniendo en cuenta las tendencias actuales, la construcción basada en código abierto del módulo Mapificación del Sistema de Prevención del Delito de Venezuela.

## CAPÍTULO 2

### CAPÍTULO 2: CARACTERÍSTICAS DEL MÓDULO MAPIFICACIÓN

#### 2. Introducción

En este capítulo se describe la propuesta del módulo Mapificación del Sistema de Prevención del Delito de Venezuela. Se presenta un modelo conceptual como alternativa a la modelación de negocio, definiendo los principales conceptos que se identificaron.

Se enumeran los requisitos funcionales y no funcionales del sistema propuesto. Se identificaron los actores y casos de uso del sistema (CUS), obteniendo finalmente un modelo de CUS.

#### 2.1 Descripción de la Solución Propuesta

El módulo Mapificación debe desarrollarse sobre una plataforma web, su implementación proporcionará al Sistema de Prevención del Delito de Venezuela de una herramienta capaz de representar sobre mapas digitales la información necesaria para contribuir en el proceso de toma de decisiones. Para dar cumplimiento a esta meta se especificaron las funcionalidades necesarias para la interacción eficiente del usuario con el sistema.

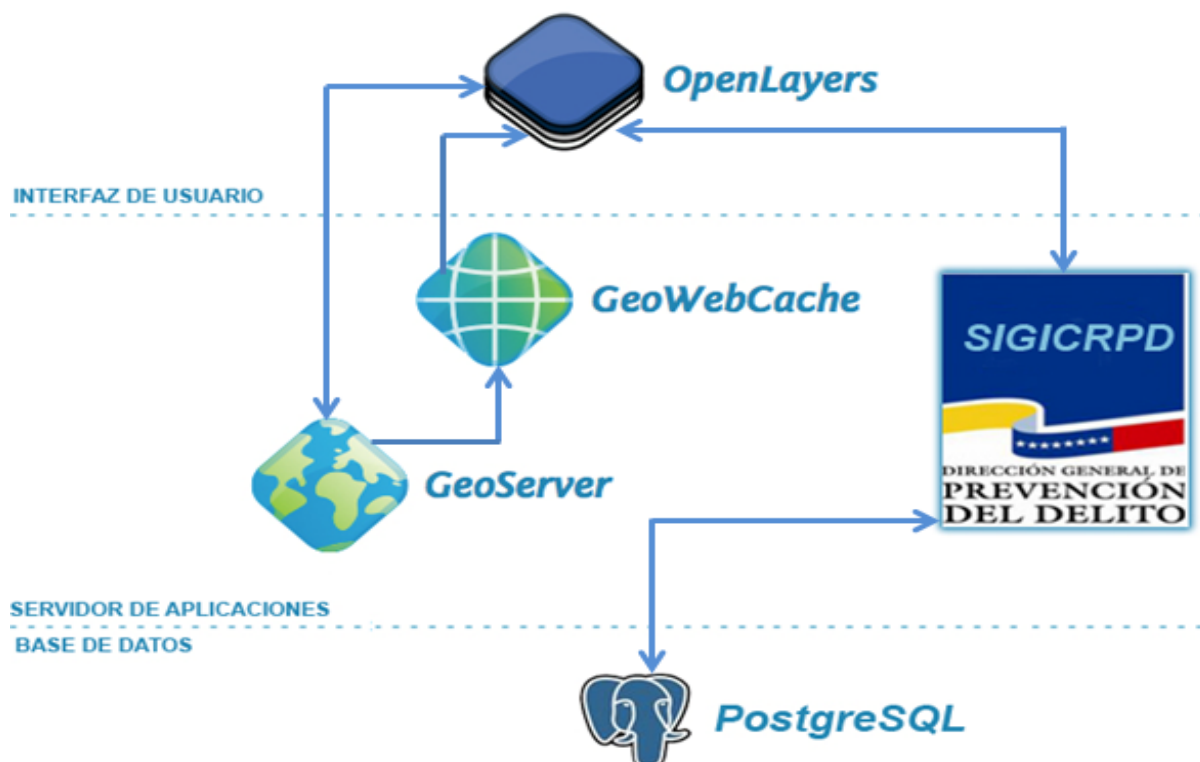


Figura 2. Concepción general de la arquitectura del módulo Mapificación.

La aplicación debe permitir acceder a la cartografía de Venezuela que se encuentra publicada en el servidor de mapas, así como al servidor de base de datos del Sistema de

## CAPÍTULO 2

Prevención del Delito de Venezuela para la obtención de los datos primarios. Inicialmente solicita las capas básicas (estados y municipios de Venezuela) para formar el mapa y se le ofrece la posibilidad al usuario de seleccionar la capa que desea visualizar.

La biblioteca OpenLayers ofrece un panel de navegación que será ubicado en la parte superior izquierda del cuadro de visualización del mapa, el cual va a permitir acercar, alejar y mover el mapa. Estas funcionalidades también podrán realizarse utilizando un teclado. Igualmente contará con varios controles como la escala que estará situada en la parte inferior del cuadro de visualización del mapa, la cuadrícula que servirá como base de apoyo en la orientación podrá observarse como una capa sobre el mapa y el selector de capas que va a permitir seleccionar que capa se desea visualizar en mapa.

El módulo Mapificación será capaz de mostrar en el mapa la información que se gestiona en los módulos pertenecientes al Sistema de Prevención del Delito de Venezuela, mediante la creación de nuevas capas que se adicionarán al mapa, capacitadas para representar los tipos de reportes solicitados, estos reportes podrán ser filtrados por la fecha en que fueron realizados mediante los calendarios que se ubicarán en el panel derecho de la aplicación.

### 2.2 Estructura del Mapa

El mapa a mostrar en el módulo Mapificación estará compuesto por las siguientes capas:

- **Capas Bases:** Capas de datos vectoriales que se solicitan a servidor de mapas GeoServer utilizando el servicio WMS, estas contendrán la información geográfica sobre los estados y los municipios de Venezuela.
- **Capa de Reportes (Vector-Point):** mostrará los reportes puntuales<sup>6</sup> del Sistema de Prevención del Delito de Venezuela y los detalles de los mismos.
- **Capa de Reportes (Vector-GML):** mostrará los reportes por niveles<sup>7</sup> del Sistema de Prevención del Delito de Venezuela, a partir de la información obtenida de la base de datos y las capas vectoriales que se solicitarán al servidor de mapas, utilizando el servicio WFS.
- **Capa de Controles:** contendrá todos los controles del mapa (cuadrícula, barra de navegación, escala, información general del mapa y el selector de capas).

---

<sup>6</sup> Reportes puntuales: reportes ubicados en una coordenada específica (Latitud, Longitud).

<sup>7</sup> Reportes por niveles: reportes que representarán con un color definido el nivel de ocurrencia de un tipo de reporte en determinadas regiones del país.



## CAPÍTULO 2

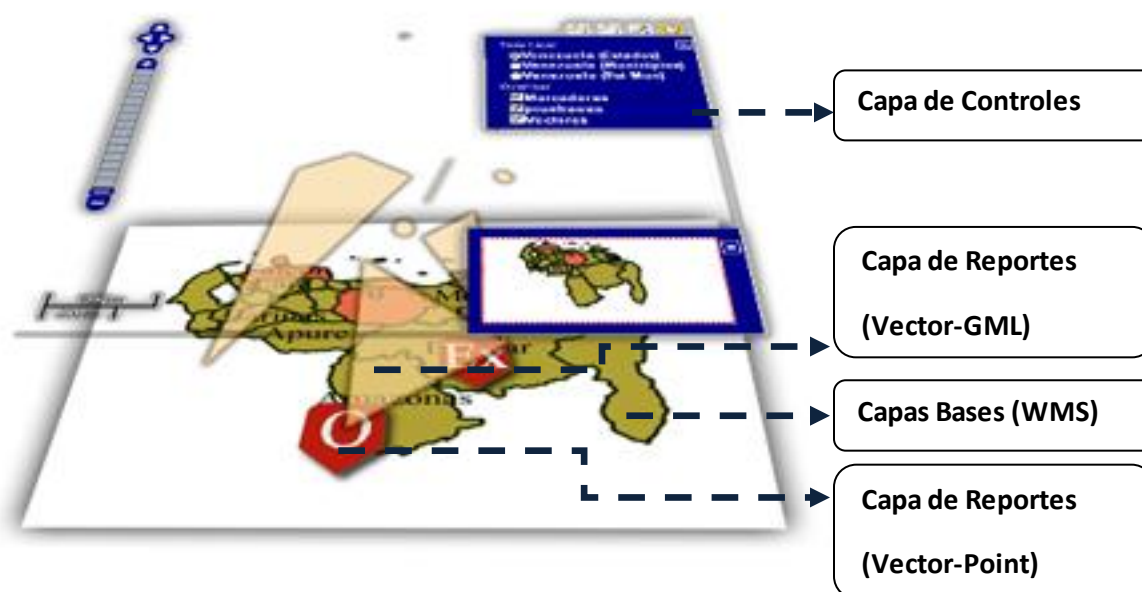


Figura 3. Estructura del Mapa.

### 2.3 Modelo Conceptual

Con el fin de lograr una mejor concepción del sistema que se quiere implementar, se propone un modelo de conceptual, que tiene como objetivo capturar y expresar el conocimiento adquirido del módulo Mapificación después de haber realizado varios análisis.

#### 2.3.1 Conceptos asociados al módulo Mapificación

**SIGICRPD:** Sistema Informático de Gestión de Información de las Coordinaciones Regionales de Prevención del Delito de Venezuela (Sistema de Prevención del Delito de Venezuela).

**UsuarioSistema:** Persona que trabaja en el SIGICRPD, a la cual le fue asignado un rol con el fin de controlar el acceso a las funcionalidades que ofrece el módulo Mapificación.

**Mapa:** Conjunto de capas y controles.

**Capa:** Es un conjunto de elementos de un mismo tipo y con los mismos atributos referenciados geográficamente (Capas Bases, Marcadores, Vectores, GML).

**Escala:** Es la relación entre las dimensiones del mapa y las dimensiones reales de la superficie que se representa.

**Leyenda:** Es la explicación de los símbolos que se muestran en el mapa, y estilos utilizados en las capas.

**Vector:** Puntos y Capas GML que van a dibujarse sobre el mapa, con el fin de representar reportes puntuales y por niveles.

## CAPÍTULO 2

**Shapefile:** Es un archivo informático que almacena las entidades geométricas (polígonos) geo-referenciadas que van a representar las capas bases.

**Polígono:** Es una entidad geométrica que almacena información geográfica de los estados y municipios de Venezuela.

**Estado:** Es una entidad territorial formada por municipios.

**Municipio:** Es una entidad territorial formada por parroquias.

**Reporte Puntual:** Es la información que se representa sobre el mapa en una coordenada (latitud, longitud) específica.

**Reporte por Nivel:** Es la información que se representa sobre el mapa dibujando los estados o municipios con un color definido según la cantidad de ocurrencia del tipo de reporte en esas entidades territoriales.

**Área:** Es un fragmento específico del Mapa que permite visualizar todos los reportes que se encuentren dentro de su perímetro.

**Control:** Herramienta diseñada para la orientación, manipulación, y navegación sobre el mapa y sus capas.

**GML:** Capa vectorial construida con un conjunto de reglas que van a especificar cuáles son las entidades territoriales que se van a representar en el mapa y el color con que serán dibujas las mismas.

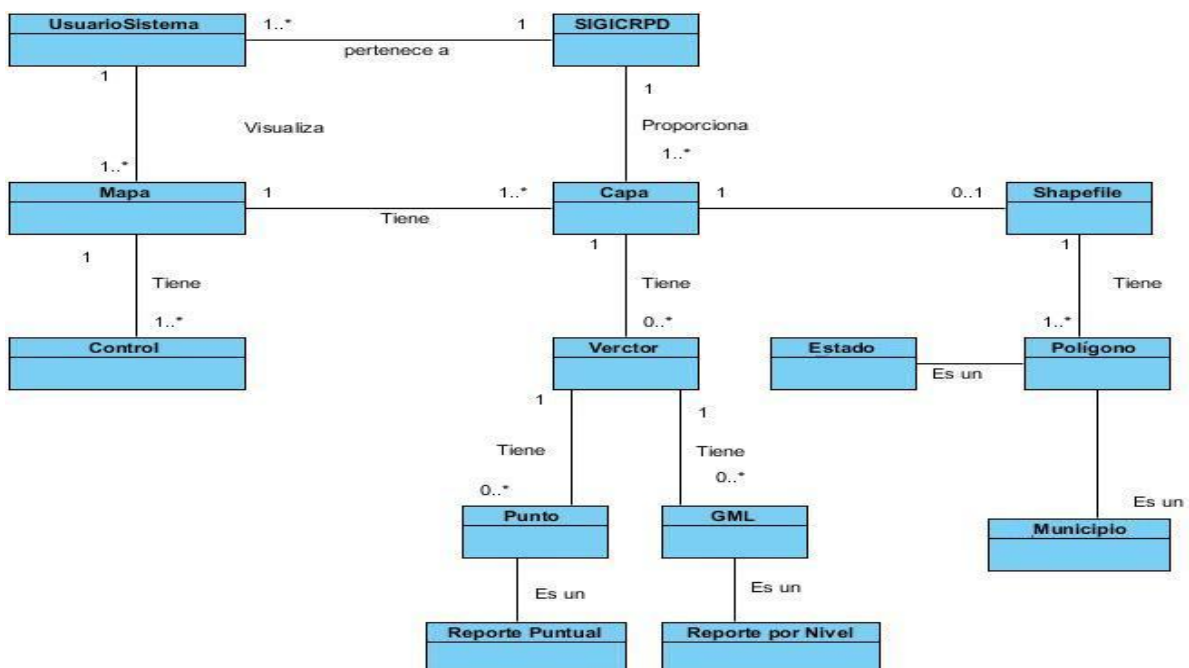


Figura 4. Modelo Conceptual del módulo Mapificación.

## CAPÍTULO 2

**Descripción del modelo conceptual:** Los usuarios del Sistema de Prevención del Delito de Venezuela que tengan permisos para acceder al módulo Mapificación podrán observar mapas digitales, estos van a estar compuestos por controles (escala, leyenda, información general del mapa, barra de navegación y selector de capas) y un conjunto de capas vectoriales que se encontrarán en un archivo con formato Shapefile (son las capas bases que se solicitarán al servidor de mapas usando el servicio WMS y representarán en el mapa los estados y municipios de Venezuela) y vector (son las capas vectoriales que se encargarán de representar la información obtenida del servidor de bases de datos del Sistema de Prevención del Delito de Venezuela sobre las capas bases del mapa mediante los reportes puntuales y por niveles).

### 2.4 Requisitos del Software

#### 2.4.1 Requisitos Funcionales

Los requisitos funcionales describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúe el sistema. (28)

##### **RF1. Cargar Configuración**

Se cargará en la aplicación los permisos a las que accederá el usuario autenticado.

##### **RF2. Crear y Visualizar Mapa**

Se crea y se visualiza el mapa con las capas que lo conforman.

##### **RF3. Acercar la Vista del Área Visible**

Visualiza el área visible del mapa a mayor escala.

##### **RF4. Alejar la Vista del Área Visible**

Visualiza el área visible del mapa a menor escala.

##### **RF5. Mover Mapa**

Permite al usuario desplazarse por el mapa en cualquier dirección.

##### **RF6. Activar Capa**

Permite al usuario seleccionar la capa del mapa que se desea visualizar.

##### **RF7. Desactivar Capa**

Permite al usuario ocultar el contenido de una o más capas visualizadas.

## CAPÍTULO 2

### **RF8. Visualizar un reporte puntual en el Mapa**

Permite al usuario mostrar un reporte en una coordenada (latitud, longitud) del mapa.

### **RF9. Visualizar Detalles de un Reporte**

Permite mostrar al usuario un resumen de la información referente al reporte que se está visualizando en el mapa.

### **RF10. Aplicar estilos a las capas que conforman el Mapa**

Visualiza el mapa con los estilos aplicados a las capas que lo conforman.

### **RF11. Visualizar un reporte por niveles en el Mapa**

Visualiza en el mapa entidades territoriales con colores definidos según la cantidad de ocurrencias de un tipo de reporte en esas entidades.

### **2.4.2 Requisitos no Funcionales**

El sistema debe ser eficiente en el campo de acción en que se utiliza, no solo es importante especificar las funcionalidades que debe desempeñar, también se tienen que definir todas las cualidades que lo hacen más atractivo al cliente, usable, rápido y confiable ya que le agregan funcionalidad.

#### **Usabilidad**

El sistema proporcionará una interfaz sencilla y atractiva, con facilidad de uso para usuarios con conocimientos mínimos de informática.

Los grupos de botones y vínculos deben estar organizados por la funcionalidad, con el objetivo de facilitar al usuario la interacción con el sistema.

Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos, en idioma español y no deben revelar información interna.

Los colores a utilizar en el desarrollo del módulo Mapificación deben ajustarse a los colores del Sistema de Prevención del Delito de Venezuela.

#### **Seguridad**

Solo podrán acceder al módulo Mapificación y hacer uso de sus funcionalidades los usuarios que estén registrados en el Sistema de Prevención del Delito de Venezuela y tengan permisos de acceso al módulo.

## CAPÍTULO 2

### Confiabilidad

El módulo Mapificación debe recuperarse ante el surgimiento de cualquier error, de no ser posible, enviar alertas al personal responsable de la administración del mismo, así como garantizar la protección de la información.

### Requerimientos del Hardware

Clientes.

- Tarjeta de red.
- 256 MB de memoria RAM mínimo.
- Microprocesador de 1GHz.

Servidores.

- Tarjeta de red.
- 4 GB de memoria RAM.
- Microprocesador de 2.5 GHz.
- 5 GB de espacio libre en el disco duro.

### Requerimientos de Software

Clientes.

- Navegador Web que cumpla con los estándares de la World Wide Web Consortium (W3C).

Servidores.

- Servidor de Bases de Datos PostgreSQL versión 8.4 con la extensión PostGIS.
- Servidor de Mapas GeoServer versión 2.1.3.
- Biblioteca JavaScript OpenLayers versión 2.11.
- Contenedor Web Apache Tomcat versión 7.0.
- Java Runtime Environment (jre) 7.0.
- Sistema Operativo CentOS 5.5.

## 2.5 Definición de los casos de uso

### 2.5.1 Definición de los actores del Sistema

Un actor es una agrupación uniforme de personas, sistemas o máquinas externas al sistema que se está desarrollando, se relaciona con éste ya que solicita sus funcionalidades. Con el módulo Mapificación solo interactúa un actor, el cual se define a continuación:

## CAPÍTULO 2

| Actor   | Descripción   |
|---------|---|
| Usuario | <ul style="list-style-type: none"><li>➤ Persona que interactúa con el sistema.</li><li>➤ Consulta la información que va a ser representada en el mapa.</li><li>➤ Utiliza todas las funcionalidades que brinda el módulo Mapificación.</li></ul> |

Tabla 1. Actores del módulo Mapificación.

### 2.5.2 Definición de los casos de uso del sistema

- Autenticar Usuario.
- Cargar Configuración.
- Realizar Navegación.
- Gestionar Capas.
- Filtrar Reporte.
- Visualizar Reporte.
- Buscar Reporte.
- Aplicar Estilo.
- Mostrar Elemento.

### 2.5.3 Diagrama de Casos de Uso

Los casos de uso representados con el color blanco en la Figura 5 pertenecen al módulo *support* del Sistema de Prevención del Delito de Venezuela y se reutilizan en el módulo Mapificación.

## CAPÍTULO 2

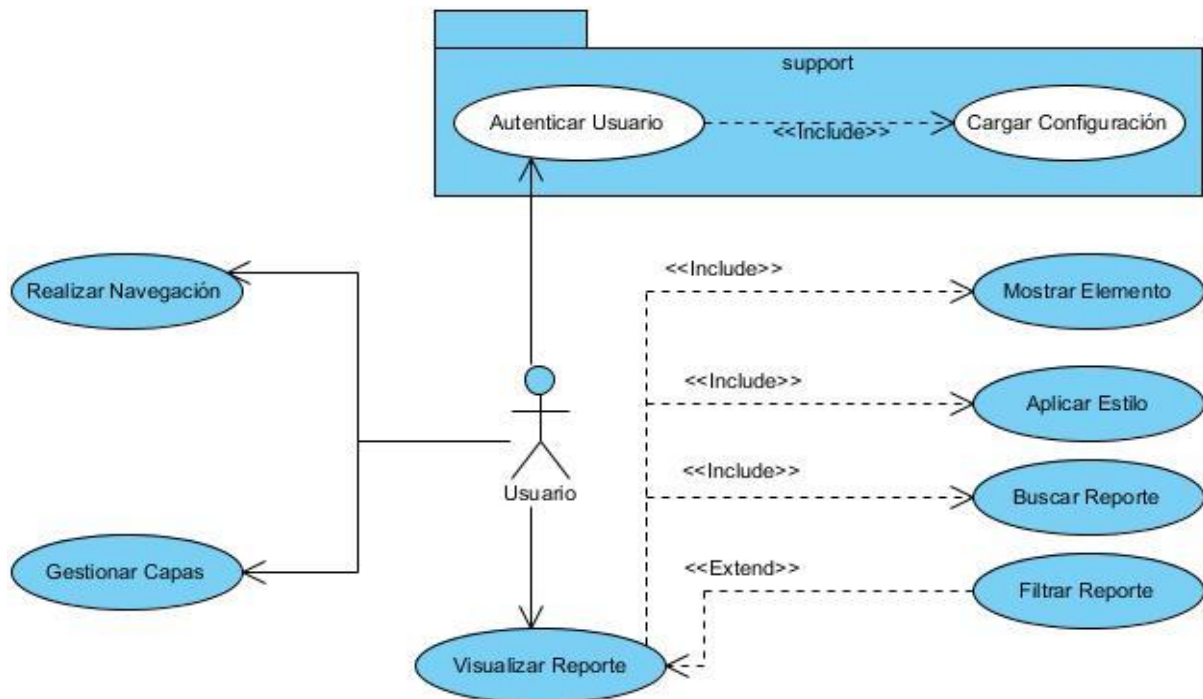


Figura 5. Diagrama de Casos de Uso.

## 2.6 Descripción de los Casos de Uso (CU)

En este epígrafe se muestra la descripción de casos de uso que se desarrollarán en el módulo Mapificación.

### 2.6.1 Descripción del CU Gestionar Capas

| Caso de Uso           | Gestionar Capas.   |
|-----------------------|--|
| <b>Objetivo</b>       | Mostrar las Capas al actor, con el propósito de que este seleccione las que se deseen visualizar en el mapa.   |
| <b>Actores</b>        | Usuario.   |
| <b>Resumen</b>        | Se muestran los nombres de las capas cargadas inicialmente, donde el usuario podrá activar y desactivar la capa que desee visualizar en el mapa. El sistema posteriormente mostrará el mapa con la configuración establecida por el usuario. |
| <b>Precondiciones</b> | El menú debe mostrarse del lado derecho del cuadro de visualización del mapa.  |
| <b>Referencias</b>    | RF6, RF7.  |

Tabla 2. Descripción del CU Gestionar Capas.

## CAPÍTULO 2

### 2.6.2 Descripción del CU Realizar Navegación

| Caso de Uso    | Realizar Navegación  |
|----------------|--|
| Objetivo       | Permitir que el actor pueda desplazarse en el mapa horizontal y verticalmente, y acercar y alejar el mapa.   |
| Actores        | Usuario.   |
| Resumen        | Se muestra la barra de navegación con la cual el usuario puede desplazarse a través del mapa, además permite acercar y alejar el mapa de manera escalar. |
| Precondiciones | La barra de navegación debe mostrarse del lado izquierdo del cuadro de visualización del mapa.   |
| Referencias    | RF3, RF4, RF5.   |

Tabla 3. Descripción del CU Realizar Navegación.

### 2.6.3 Descripción del CU Visualizar Reporte

| Caso de Uso    | Visualizar Reporte.  |
|----------------|--|
| Objetivo       | Permitir al actor visualizar reportes.   |
| Actores        | Usuario.   |
| Resumen        | Permite al usuario consultar y visualizar los reportes, puede visualizar los mismos de forma puntual (Coordenada (latitud, longitud) en el mapa) y aplicándole estilos a las capas del mapa. |
| Precondiciones | Debe existir algún reporte en la base de datos del Sistema de Prevención del Delito de Venezuela.  |
| Referencias    | RF11, RF10, RF8, RF9   |

Tabla 4. Descripción del CU Visualizar Reporte.

## 2.7 Conclusiones

La descripción de la solución propuesta ofrece una mejor comprensión del módulo que se va a implementar. Con la elaboración del modelo conceptual se brinda una visión más clara de los componentes y los conceptos asociados al módulo Mapificación, así como las relaciones entre estos. Con la creación del diagrama y la descripción de casos de uso se



## CAPÍTULO 2

especifican las funcionalidades que debe cumplir el módulo. Además se especificó que las capas bases que conforman el mapa se encuentran en formato Shapefile, el cual posibilita el almacenamiento y publicación de entidades territoriales (estados y municipios) de Venezuela en el servidor de mapas GeoServer.

## CAPÍTULO 3

### CAPÍTULO 3: CONSTRUCCIÓN DEL MÓDULO MAPIFICACIÓN

#### 3. Introducción

En el presente capítulo se abordarán todos los aspectos relacionados con el diseño e implementación del módulo Mapificación. Se muestran los diagramas de: clases del análisis, colaboración, clases del diseño del CU Visualizar Reporte, y el de clases entidades del dominio.

Se explica la arquitectura del Sistema de Prevención del Delito de Venezuela y los patrones de diseño que se utilizaron en la construcción del módulo Mapificación. Se presentan los diagramas con los componentes principales que conforman el módulo Mapificación. Se define el modelo de despliegue especificando los protocolos de comunicación existente entre los nodos.

#### 3.1 Arquitectura base del Sistema de Prevención del Delito de Venezuela

La arquitectura de este sistema fue definida en los inicios de su creación, se precisó que fuera desarrollado sobre plataforma web, utilizando Spring como framework base. Esta arquitectura se encuentra organizada desde dos enfoques (horizontal y vertical) los cuales son detallados a continuación.

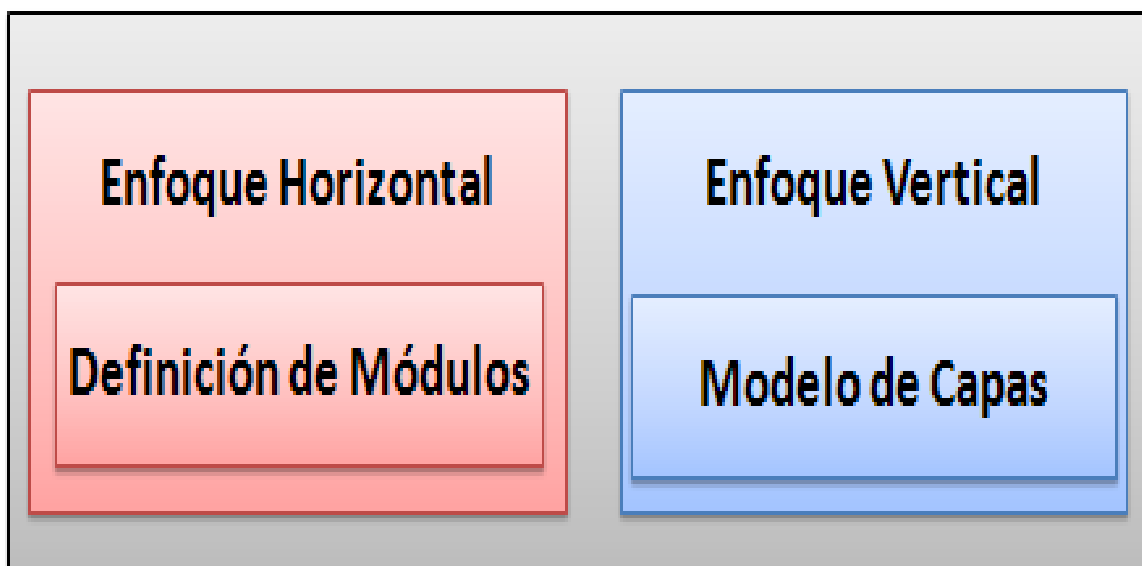


Figura 6. Enfoques de la Arquitectura del Sistema de Prevención del Delito de Venezuela.

## CAPÍTULO 3

### 3.1.2 Enfoque Horizontal

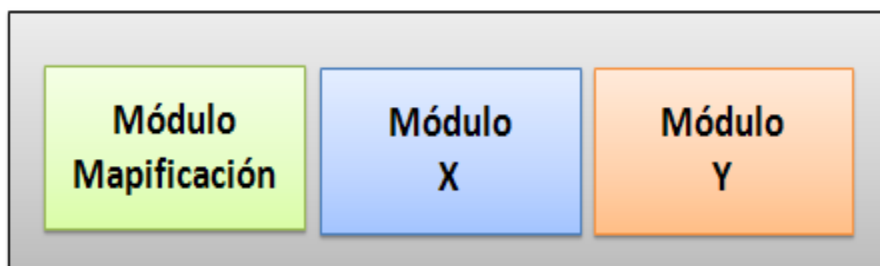


Figura 7. Enfoque Horizontal de la Arquitectura del Sistema de Prevención del Delito de Venezuela.

El Sistema de Prevención del Delito de Venezuela está compuesto por módulos, estos constituyen estructuras atómicas donde recae la implementación de las funcionalidades que debe cumplir el sistema.

### 3.1.3 Enfoque Vertical

Los módulos que conforman el Sistema de Prevención del Delito de Venezuela responden a un modelo multicapas, en el que cada capa posee funcionalidades y objetivos precisos. Los elementos de una capa sólo pueden hacer referencia a elementos en capas inmediatamente inferiores, este aspecto es muy importante porque reduce las dependencias, ya que las capas más bajas no conocen ningún elemento que posea las capas superiores a ella.

La distribución de las capas y su interacción se pueden evidenciar en la Figura 8. Las flechas representan la orientación de la dependencia y se muestra la relación de todas las capas a las entidades del dominio de un módulo.

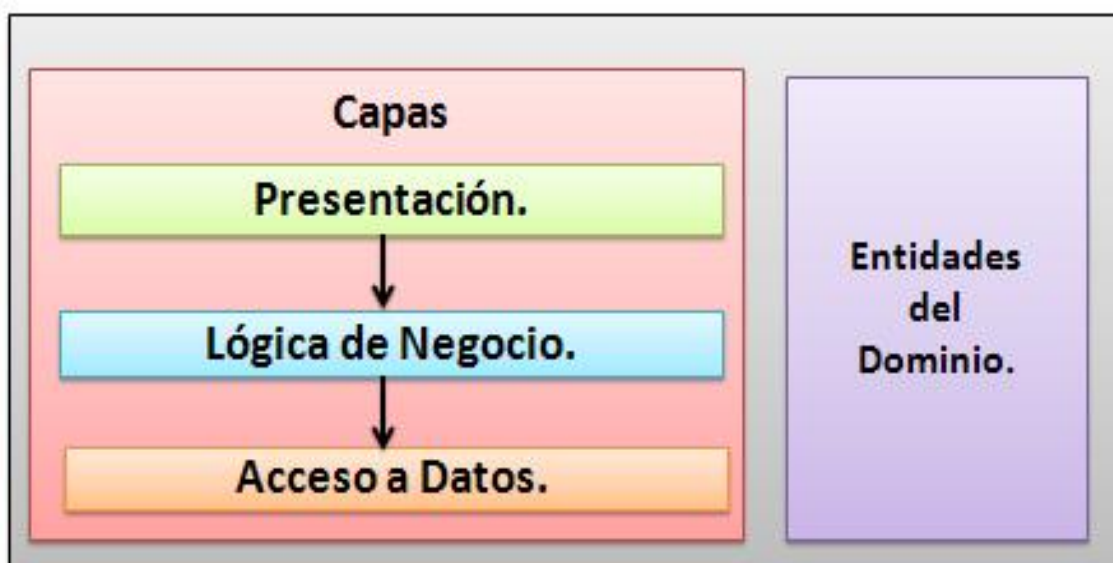


Figura 8. Modelos de Capas de un módulo del Sistema de Prevención del Delito de Venezuela.

## CAPÍTULO 3

A continuación una breve explicación de las capas que componen al estilo arquitectónico

### Arquitectura en Capas:

- **Capa de Presentación:** La capa de presentación define e implementa todo lo concerniente con la interfaz gráfica de usuario. Es el lugar donde se definen las peticiones que el usuario puede realizar sobre el sistema, los controladores que manejan el flujo web y la comunicación con las interfaces de la capa de negocio, un formulario es un ejemplo de una capa de presentación. La capa de presentación debe garantizar la navegabilidad del sistema, el formateo de los datos de salida, la validación de los datos de entrada y la construcción de la interfaz gráfica de usuario.

Es importante destacar que en esta capa va a estar presente el patrón arquitectónico Modelo-Vista-Controlador (MVC) ya que el framework base del Sistema de Prevención del Delito de Venezuela lo utiliza en su funcionamiento.

- **Capa Lógica de Negocio:** Es la parte donde se encuentra la lógica del negocio del módulo. Es la capa que toma las peticiones del usuario a través de la capa de presentación y se encarga de darle curso. Tiene la responsabilidad de definir e implementar las funcionalidades que responden directamente a los requisitos, de manera que garantice la integridad del sistema y de los datos.
- **Capa de Acceso a Datos:** Esta capa es la encargada de recobrar y persistir información desde y hacia la base de datos y de la comunicación con el gestor de base de datos. Contiene los objetos que encierran la lógica de acceso a datos (DAO) e interfaces brindadas para ser accedida desde la capa de negocio.

Las implementaciones de los DAOs extienden de clases de soporte del framework Spring para el uso de este patrón, empleando JPA con el soporte del framework Hibernate, mientras que las interfaces se mantienen independientes de Spring e Hibernate.

Las clases entidades del dominio pueden ser utilizadas desde cualquier capa.

### 3.2 Diagrama de Clases del Análisis

Los diagramas de clases del análisis son artefactos en los que se representan los conceptos en un dominio de un determinado problema. En la Figura 9 se muestra el diagrama de clases del análisis del CU Visualizar Reporte.

El diagrama de clases del análisis está compuesto por clases Entidades, Interfaces y Controladoras.

- **Clase Entidad:** Modela información que posee larga vida y que es a menudo persistente.

## CAPÍTULO 3

- **Clase Interfaz:** Modela la interacción entre el sistema y sus actores.
- **Clase Controladora:** Coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

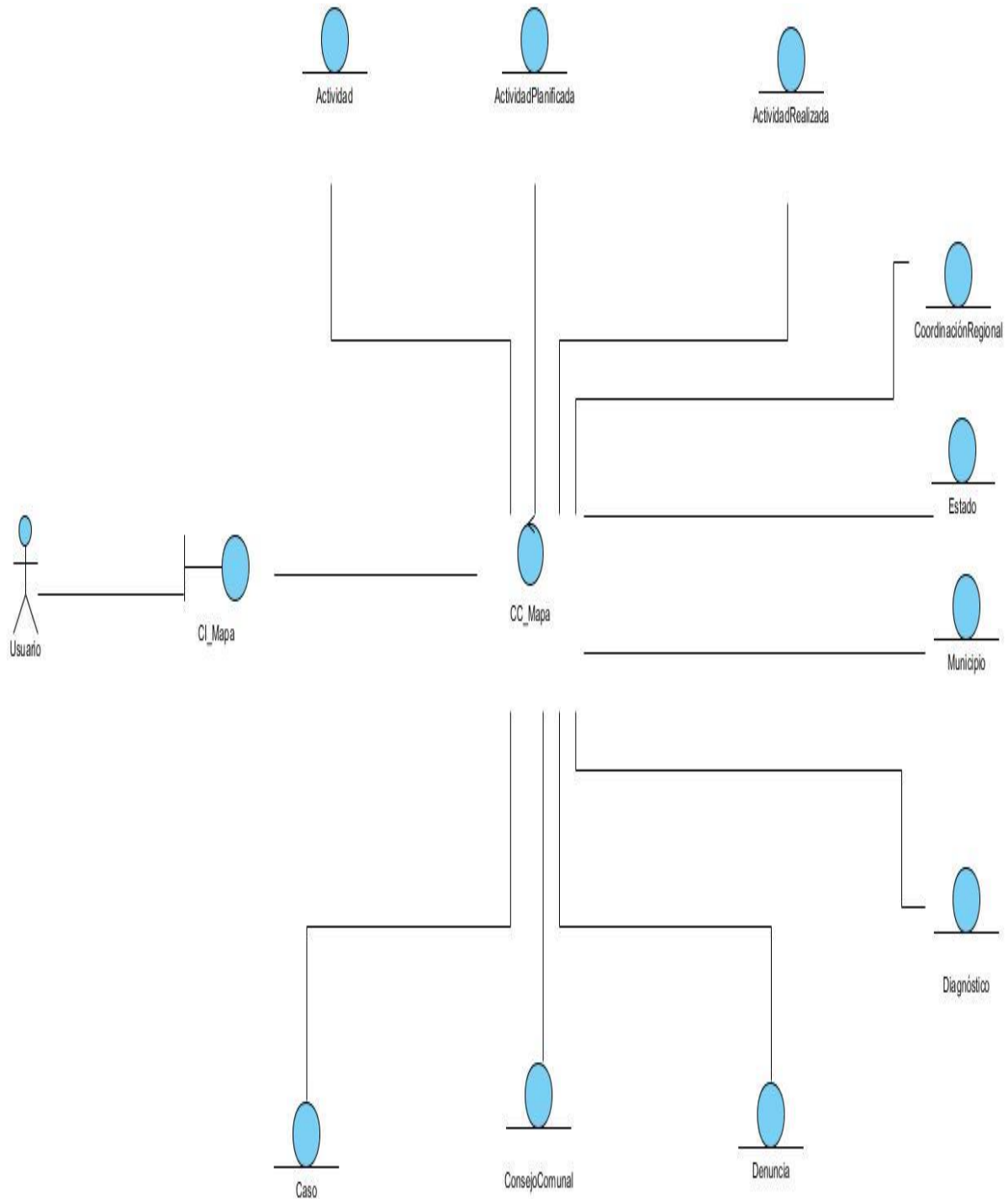


Figura 9. Diagrama de Clases del Análisis del CU Visualizar Reporte.

## CAPÍTULO 3

### 3.3 Diagrama de colaboración

Los diagramas de colaboración son diagramas de interacción que describen la forma en que colaboran un conjunto de objetos para cierto comportamiento.

A continuación se muestra en la Figura 10 el diagrama de colaboración del CU Visualizar Reporte. Este diagrama capta el comportamiento del CU y los representa mediante un número de objetos y los mensajes que se envían entre ellos.

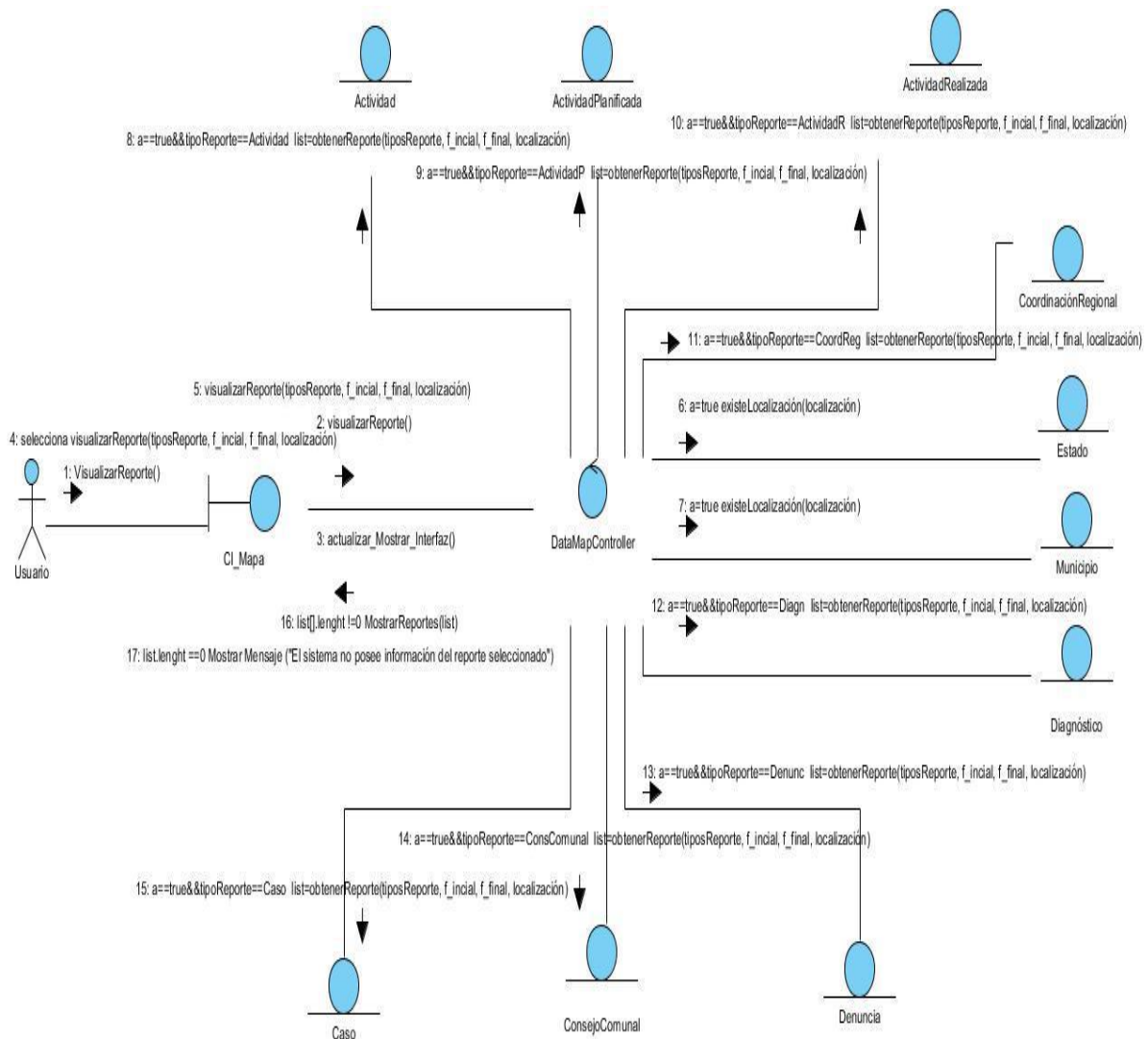


Figura 10. Diagrama de Colaboración del Análisis del CU Visualizar Reporte.

### 3.4 Patrones de Diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. (29). Ofrecen una solución ya probada y documentada a problemas generales del diseño de un software, en un determinado contexto, y encierran la experiencia que han alcanzado otras personas en la solución de problemas comunes.

## CAPÍTULO 3

El Sistema de Prevención del Delito de Venezuela se caracteriza por el uso de varios framework, lo que permite ahorrar tiempo en el diseño e implementación del sistema a desarrollar, implementando numerosos patrones de diseño.

Los patrones de diseño utilizados para la implementación del módulo Mapificación son:

- *Abstract Factory (Fábrica Abstracta)*: Crea diferentes familias de objetos o que dependen entre sí, sin especificar en sus clases concretas. Spring implementa este patrón encargándose de la creación de los objetos de la aplicación (DAOs, servicios, fachadas y controladores) y de la resolución de las dependencias que existen entre ellos. (30).
- *Singleton (Instancia única)*: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella, restringiendo la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. En el sistema se utiliza para crear una única instancia de cada clase de acceso a datos, servicios, fachadas y controladores. (31).
- *Facade (Fachada)*: Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Crea una interfaz de alto nivel que hace posible que el subsistema sea más fácil de usar, ya que funciona de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. (29).

En el sistema cada módulo define una fachada, donde se ubican todas las funcionalidades implementadas por este. De esa manera para implementar la capa web no es necesario conocer qué servicio específico brinda cada funcionalidad, sino que los controladores acceden a la fachada como único punto de acceso a las funcionalidades del módulo. La implementación de la fachada no contiene lógica de aplicación, sino que delega en los servicios la implementación real de la lógica.

- *Data Access Object (DAO)*: Proporciona una capa independiente en donde centraliza todo el acceso a datos, de forma tal que los aísla del resto de la aplicación. (29) Sus principales beneficios son: reducir la complejidad de los objetos del negocio abstrayéndolos de la implementación real de la comunicación con la fuente de datos y permitir una migración más fácil de fuente de datos.

En el caso del módulo de Mapificación se crean DAOs como ReportesDAO, que es responsable de ejecutar las consultas a la base de datos necesarias para la obtención de los datos a mostrar sobre los mapas.

## CAPÍTULO 3

### 3.5 Diagrama de Clases del Diseño

Los diagramas de clases del diseño son artefactos que genera la metodología RUP, estos especifican la estructura de clases de un sistema, así como sus relaciones. Definen de forma correcta las relaciones de dependencia, generalización y asociación de las clases que constituyen el sistema.

En la Figura 11 se representa el diagrama de clases del diseño para el caso de uso Visualizar Reporte, solo se muestra el diagrama de este CU porque es el que mayor grado de complejidad posee, ya que los CU restantes no se relacionan con las clases del módulo Mapificación, solo se relacionan con la biblioteca OpenLayers y con el servidor de mapas GeoServer.

En este CU el sistema muestra en la interfaz del módulo Mapificación el mapa de Venezuela con las capas que lo conforman. El mapa se muestra utilizando la biblioteca OpenLayers que es la encargada de solicitarle al servidor de mapas las cartografías empleando para ello las especificaciones WMS, WFS y WCS. El usuario solicita al sistema visualizar reportes, el sistema muestra los tipos de reportes que puede visualizar, el usuario selecciona los tipos de reportes que desea representar sobre el mapa completo o en un estado específico, el sistema busca en el servidor de bases de datos los reportes existentes y los muestra en la localización antes definida. En los Anexos se encuentra la descripción de las clases entidades con sus atributos.

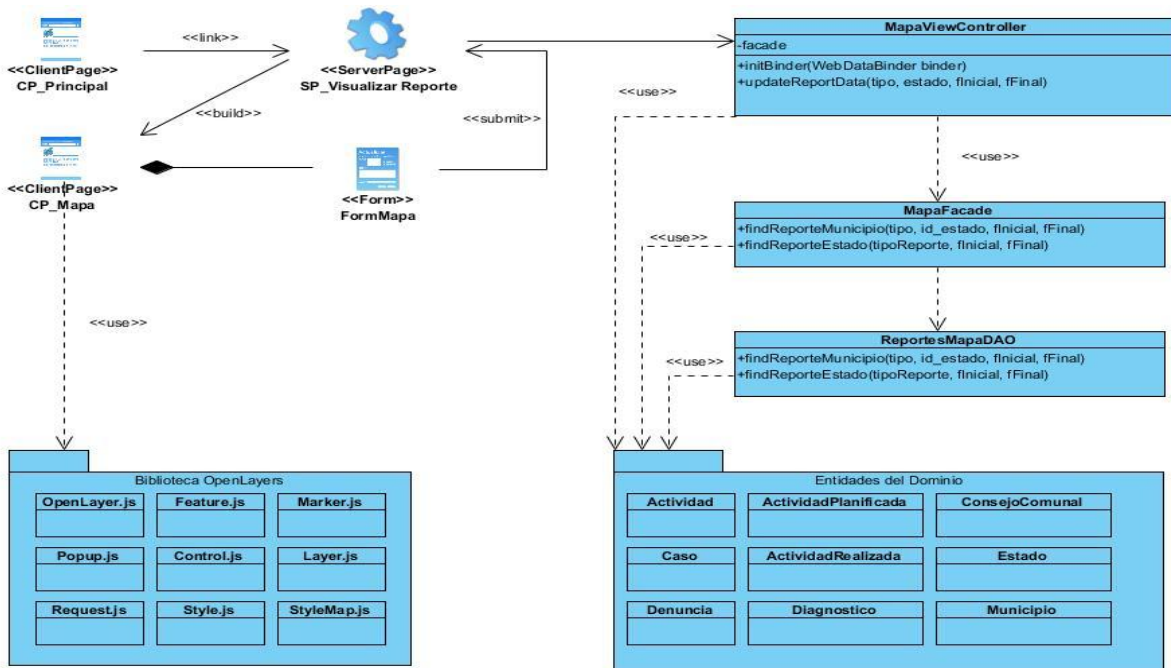


Figura 11. Diagrama de Clases del Diseño. CU Visualizar Reporte.



## CAPÍTULO 3

### 3.6 Diagrama de Clases Entidades del Dominio

A continuación se muestra el Diagrama de Clases Entidades del Dominio, en los Anexos se encuentra una descripción de estas clases y sus atributos.

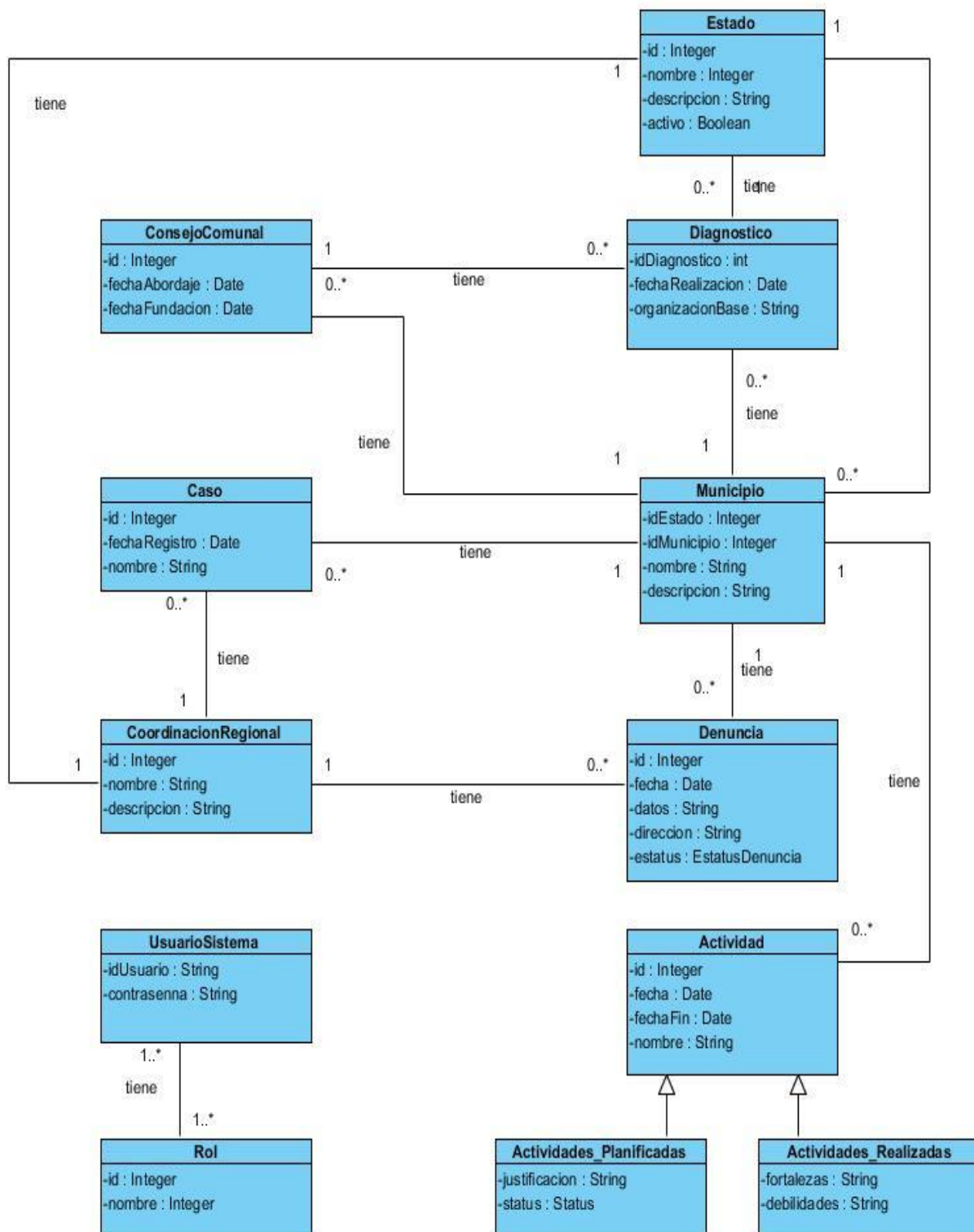


Figura 12. Diagrama de Clases Entidades del Dominio.

## CAPÍTULO 3

### 3.7 Modelo de Datos

A continuación se muestra en la Figura 13 el modelo de datos del módulo Mapificación el cual ofrece una descripción abstracta sobre la representación de los datos en un SGBD. Los componentes que lo conforman, son entidades existentes en el Sistema de Prevención del Delito de Venezuela y tablas espaciales que contienen información geo-referencia de los estados y municipios de Venezuela. Este modelo contienen a su vez, características propias de estos objetos y la forma en que se relacionan entre sí.

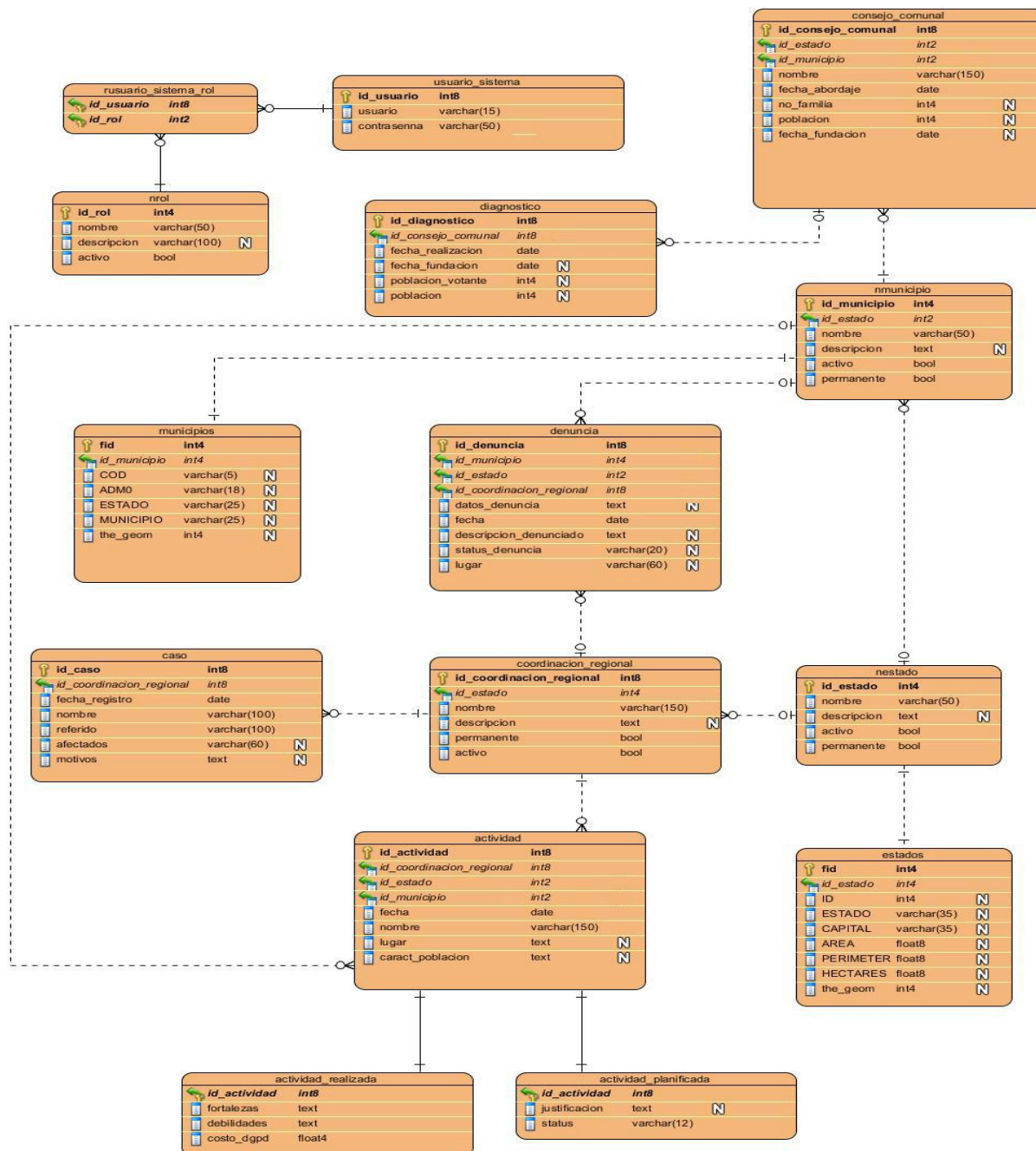


Figura 13. Modelo de Datos del módulo Mapificación.

## CAPÍTULO 3

### 3.8 Modelo de Implementación

El modelo de implementación representa cómo los elementos del modelo de diseño se implementan en términos de componentes. De igual forma describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y la relación existentes entre ellos.

A continuación se muestra en la Figura 14 el diagrama de componentes correspondiente al módulo Mapificación.

#### 3.8.1 Diagrama de Componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Exponen la organización y las dependencias lógicas entre un conjunto de componentes de software, sean éstos: código fuente, librerías, binarios, ejecutables. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación.

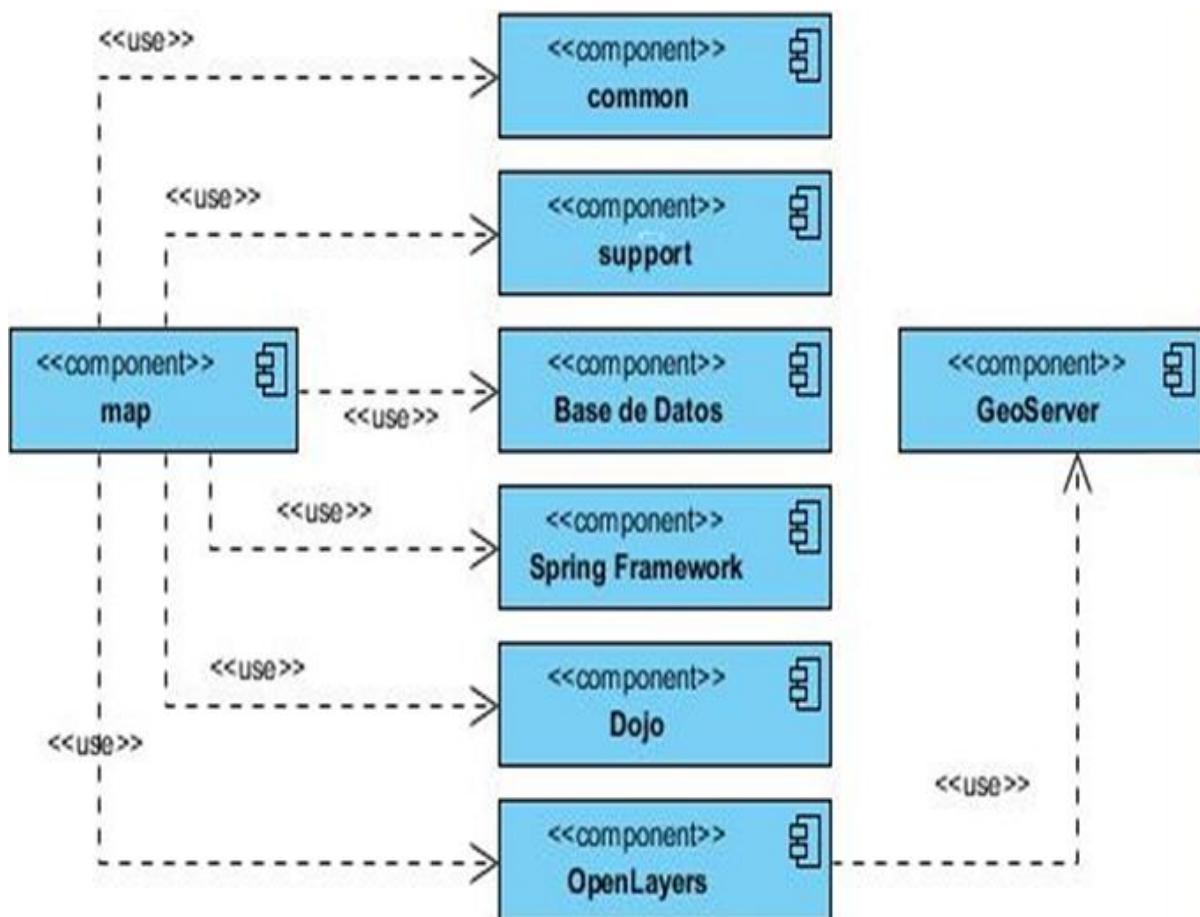


Figura 14. Diagrama de Componentes del módulo Mapificación.

## CAPÍTULO 3

### Descripción de los Componentes

- **Componente map:** El componente map es el módulo Mapificación del Sistema de Prevención del Delito de Venezuela, este componente contiene el código fuente correspondiente a la capa de presentación, lógica de negocio y al acceso a datos de dicho módulo.
- **Componente common:** El componente common es un módulo del Sistema de Prevención del Delito de Venezuela el cual se encarga de construir el *datasource* de la aplicación. Desde el módulo Mapificación se ejecutan consultas que necesitan el *datasource* construido por este componente.
- **Componente support:** El componente support es un módulo del Sistema de Prevención del Delito de Venezuela. El módulo Mapificación lo utiliza para convertir los datos de lenguaje java a JavaScript, con el fin de poder mostrar la información en las vistas. En este componente también están definidas clases e interfaces que se utilizan en todos los demás módulos de la aplicación como la interfaz e implementación base de los Dao y AbstractDao, la interfaz e implementación base de los objetos de la capa de negocio (Manager y AbstractManager), la configuración de la seguridad del sistema y algunas clases de utilidades como FTPConnection.
- **Componente Base de Datos:** Este componente es la base de datos del Sistema de Prevención del Delito de Venezuela, contiene tablas, vistas y funciones que van a ser utilizadas por el módulo Mapificación con el objetivo de representar reportes sobre un mapa digital. Ejemplos de funciones utilizadas.
  - Centroid: Calcula el centro de un polígono.
  - St\_x: Devuelve la coordenada X de un punto.
  - St\_y: Devuelve la coordenada X de un punto.
- **Componente Spring Framework:** El componente Spring es un framework de aplicación que está presente en todas las capas que conforman el Sistema de Prevención del Delito de Venezuela.
- **Componente Dojo:** Este componente es utilizado por el módulo Mapificación para generar componentes gráficos con el objetivo de elevar la calidad de la IU y a la vez aumentar la usabilidad.

## CAPÍTULO 3

- **Componente OpenLayers:** El componente OpenLayers es una librería JavaScript encargada de crear mapas dinámicos a partir de la información obtenida del servidor de mapas y de la base de datos del Sistema de Prevención del Delito de Venezuela.
- **Componente GeoServer:** El componente GeoServer es el servidor de mapas, es el encargado de brindar la cartografía de Venezuela que solicite la biblioteca OpenLayers utilizando los servicios WMS y WFS.

### 3.9 Modelo de Despliegue

El Modelo de despliegue contiene los nodos que forman la topología hardware sobre la que se ejecuta el módulo Mapificación, el software necesario para su funcionamiento y los protocolos de comunicación. Al módulo Mapificación del Sistema de Prevención del Delito de Venezuela se puede acceder desde un cliente utilizando un navegador web que cumpla con los estándares definidos por la W3C. Este navegador envía o recibe información desde el Sistema de Prevención del Delito de Venezuela que se encuentra alojado junto a la biblioteca OpenLayers y el servidor de mapas GeoServer en el contenedor web Apache Tomcat.

A continuación en la Figura 15 se muestra el diagrama de componentes correspondiente al módulo Mapificación.

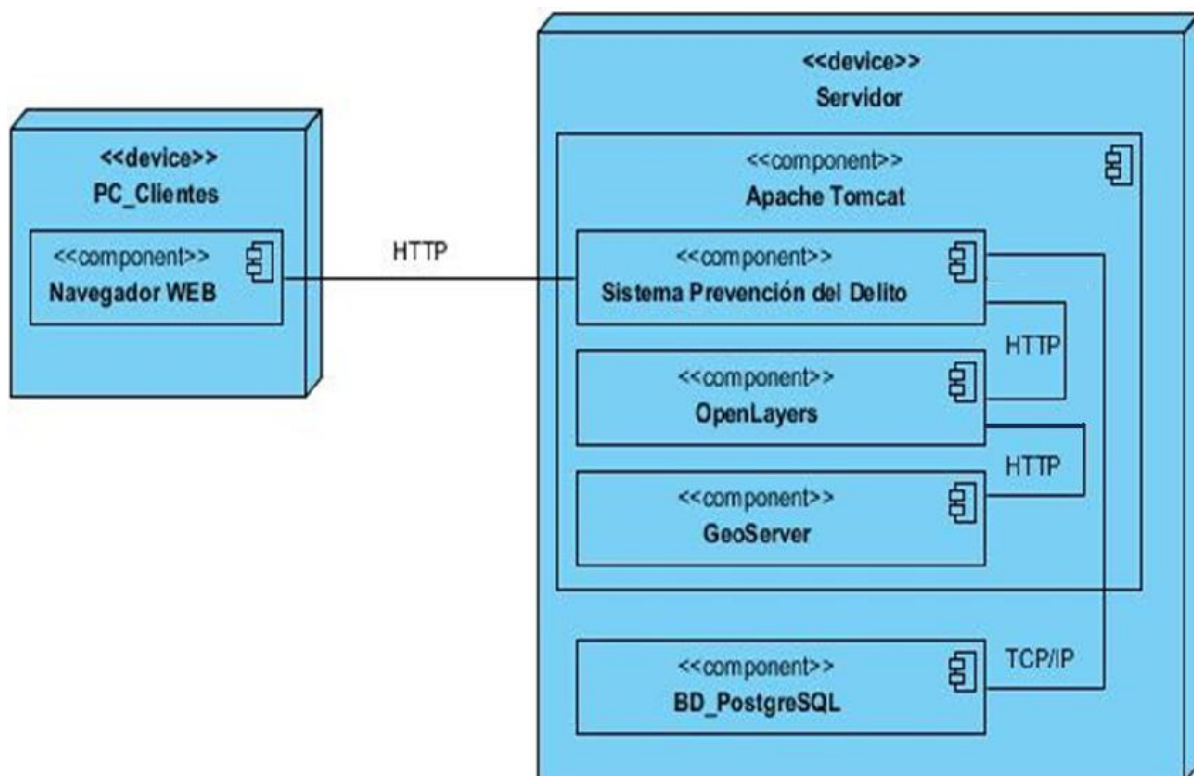


Figura 15. Diagrama de Despliegue del módulo Mapificación.

## CAPÍTULO 3

### 3.9.1 Descripción de los Protocolos de Comunicación

Se utilizaron los protocolos de comunicación TCP/IP y HTTP debido a que el módulo Mapificación se desarrolla sobre tecnología Web siguiendo los estándares para la misma.

- **Familia de Protocolos TCP/IP.** Se utiliza para enlazar computadoras que usan sistemas operativos similares o diferentes, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN). En el caso del Sistema de Prevención de Delito se utiliza para interconectarlo con el servidor de bases de datos.
- **Protocolo HTTP.** El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW), HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (cliente, servidor, proxy) para comunicarse. En el Sistema de Prevención del Delito de Venezuela se utiliza este protocolo para el acceso de los clientes web a los servicios que brinda la aplicación.

### 3.10 Conclusiones

En este capítulo se llevó a cabo todo el proceso relacionado con el diseño e implementación del módulo Mapificación. Con la descripción de la arquitectura del Sistema de Prevención del Delito de Venezuela, quedó especificado el lugar que ocupa el módulo Mapificación dentro de este sistema. Se realizó una breve descripción de los patrones de diseño y arquitectónicos utilizados en el desarrollo del módulo.

Se describió el diseño del módulo Mapificación a través de los diagramas de clases del análisis y del diseño. Se crearon los diagramas de componentes y despliegue correspondientes al modelo de implementación, quedando definida la distribución física y lógica de la arquitectura del sistema y sus conexiones.

## CAPÍTULO 4: PRUEBAS AL SISTEMA PROPUESTO

### 4. Introducción

En el presente capítulo se exponen todos los aspectos relacionados con las pruebas de caja negra y pruebas de caja blanca efectuadas al módulo Mapificación, con el objetivo de identificar errores.

#### 4.1 Pruebas de Caja Blanca

Las pruebas de Caja Blanca permiten asegurar que la operación interna se ajusta a las especificaciones del software, y que todos los componentes internos se han probado correctamente.

Al módulo Mapificación se le aplicaron pruebas unitarias con el objetivo de encontrar errores en el código, para la realización de las mismas se utilizó la herramienta JUnit, ya que se integra fácilmente con el entorno de desarrollo y permite elaborar casos de pruebas.

Se crearon casos de pruebas y fueron aplicados a la clase “**ReportesMapaDAO**” con el objetivo de encontrar errores en las consultas utilizadas para mostrar la información en el mapa a nivel de estados y municipios. A continuación en la Figura 16 se muestra el resultado que arroja la herramienta JUnit al efectuar el caso de prueba diseñado para verificar el funcionamiento de la consulta que devuelve la cantidad de denuncias registradas en la base de datos del Sistema de Prevención de Delito de Venezuela en un rango de fecha establecido.

```

21 public class ReportesMapaDAOTest {
22
23     private ReportesMapaDAO reportesMapaDAO;
24
25     private SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
26
27     @Autowired
28     public void setReportesMapaDAO(ReportesMapaDAO reportesMapaDAO) {
29         this.reportesMapaDAO = reportesMapaDAO;
30     }
31
32     /**
33      * Método para probar la consulta a niv
34      *
35      * @throws Exception
36      */
37     @Test
38     public void testReporteDenuncias() throws Exception {
39         // probar que hay 1 denuncia en el mes de marzo
40         List<Object[]> list = reportesMapaDAO.findReporteEstado("denuncia",
41             "fecha", dateFormat.parse("1/3/2013"), dateFormat.parse("31/3/2013"));
42         assertEquals(1, list.size());
43         // probar que hay 1 denuncia en el mes de abril
44         list = reportesMapaDAO.findReporteEstado("denuncia",
45             "fecha", dateFormat.parse("1/4/2013"), dateFormat.parse("3/4/2013"));
46         assertEquals(4, list.size());
47         // probar que hay 9 denuncias en el mes de mayo
48         list = reportesMapaDAO.findReporteEstado("denuncia",
49             "fecha", dateFormat.parse("1/5/2013"), dateFormat.parse("31/5/2013"));
50         assertEquals(9, list.size());
51     }
52 }
53

```

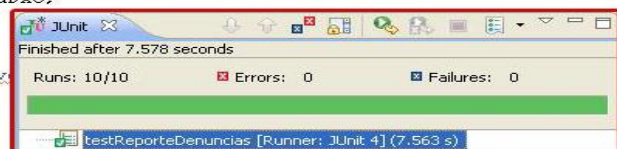


Figura 16. Resultado de la aplicación de un caso de prueba utilizando la herramienta JUnit.

## CAPÍTULO 4

Al aplicar los casos de pruebas al módulo no se detectaron deficiencias en las consultas realizadas a la base de datos.

### 4.2 Pruebas de Caja Negra

Las pruebas de Caja Negra permiten comprobar que cada función del sistema desarrollado es operativa, además se pueden identificar errores que posea el mismo. A continuación se muestran los casos de prueba de caja negra aplicadas a los CU del módulo Mapificación.

#### 4.2.1 Pruebas de Caja Negra del CU Realizar Navegación

| Nombre del Caso de Uso: Realizar Navegación   |  |  |
|---|--|--|
| Entrada   | Resultado  | Condiciones  |
| EC 1: El actor selecciona hacia la posición (Izquierda, Derecha, Arriba, Abajo) hacia la que desea mover el mapa, también puede acercar o alejar el mapa. | El sistema mueve, acerca o aleja el mapa, según la opción seleccionada por el actor. | El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas. |

Tabla 5. Pruebas de Caja Negra del CU Realizar Navegación.

#### 4.2.2 Pruebas de Caja Negra del CU Gestionar Capas

| Nombre del Caso de Uso: Gestionar Capas.  |  |  |
|---|--|--|
| Entrada   | Resultado  | Condiciones  |
| EC 1: El actor selecciona la(s) capa(s) del mapa que desea visualizar o deshabilitar. | El sistema visualiza o deshabilita la(s) capa(s) del mapa que el actor seleccionó. | El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas. |

Tabla 6. Pruebas de Caja Negra del CU Gestionar Capas.



## CAPÍTULO 4

## 4.2.3 Pruebas de Caja Negra del CU Visualizar Reporte

| Nombre del Caso de Uso:  | Visualizar Reporte  |  |
|--|---|--|
| Entrada  | Resultado   | Condiciones  |
| EC 1: El actor selecciona la opción visualizar reportes.   | El sistema muestra una lista con los tipos de reportes que se pueden visualizar, muestra los campos correspondientes para filtrar el reporte por estado y por un rango de fechas.   | El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas. |
| EC 1.1: El actor selecciona el tipo de reporte “Denuncias” para visualizarlo, sin aplicar ningún filtro. | <p>El sistema busca en la base de datos la cantidad existente de “Denuncias” en cada estado y muestra en el mapa a nivel de estados creando un marcador en el centro del estado en el cual se ofrece detalles del reporte (Estado, Cantidad de Denuncias), también se dibujan las entidades territoriales (en este caso los estados) con colores definidos según la cantidad de denuncias que tenga el estado.</p> <p>El sistema crea dos capas, las capas “Reporte de Denuncias” y “Reporte de Denuncias por Niveles”.</p> | El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas. |
| EC 1.2: El actor selecciona el tipo de reporte “Denuncias” que desea                                     | El sistema busca en la base de datos la cantidad existente de “Denuncias” efectuadas entre el   | El actor debe estar autenticado y las capas bases del Mapa deben                 |

## CAPÍTULO 4

|  |   |   |
|--|---|---|
| <p>visualizar en el mapa, selecciona filtrarlo por un rango de fecha:</p> <p>Fecha inicial: 10/01/2013</p> <p>Fecha final: 21/05/2013</p>                    | <p>10/01/2013 y el 21/05/2013 de cada estado y muestra en el mapa a nivel de estados creando un marcador en el centro del estado en el cual se ofrece detalles del reporte (Estado, Cantidad de Denuncias), también se dibujan las entidades territoriales (en este caso los estados) con colores definidos según la cantidad de “Denuncias” que tenga el estado.</p> <p>El sistema crea dos capas, las capas “Reporte de Denuncias” y “Reporte de Denuncias por Niveles”.</p>  | <p>estar cargadas.</p>  |
| <p>EC 1.3 : El actor selecciona el tipo de reporte “Denuncias” que desea visualizar en el mapa, selecciona filtrarlo un por Estado:</p> <p>Estado: Sucre</p> | <p>El sistema busca en la base de datos la cantidad existente de “Denuncias” pertenecientes a los municipios del estado Sucre y muestra en el mapa a nivel de municipios creando un marcador en el centro de cada municipio en el cual se ofrece detalles del reporte (Estado, Municipio, Cantidad de Denuncias), también se dibujan las entidades territoriales (en este caso los municipios) con colores definidos según la cantidad de “Denuncias” que tenga el municipio.</p> <p>El sistema crea dos capas, las</p> | <p>El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas.</p> |

## CAPÍTULO 4

|  |   |   |
|--|---|---|
|  | capas “Reporte de Denuncias” y “Reporte de Denuncias por Niveles”.  |   |
| <p>EC 1.4 : El actor selecciona el tipo de reporte “Denuncias” que desea visualizar en el mapa, selecciona filtrarlo por un Estado y por un rango de fecha:</p> <p>Estado: Sucre</p> <p>Fecha inicial: 10/01/2013</p> <p>Fecha final: 21/05/2013</p> | <p>El sistema busca en la base de datos la cantidad existente de “Denuncias” realizadas entre el 10/01/2013 y el 21/05/2013 pertenecientes al Estado “Sucre” y muestra en el mapa a nivel de municipios creando un marcador en el centro de cada municipio en el cual se ofrece detalles del reporte (Estado, Municipio, Cantidad de Denuncias), también se dibujan las entidades territoriales (en este caso los municipios) con colores definidos según la cantidad de “Denuncias” que tenga el municipio.</p> <p>El sistema crea dos capas, las capas “Reporte de Denuncias” y “Reporte de Denuncias por Niveles”.</p> | <p>El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas.</p> |
| <p>EC 2. El actor selecciona la opción visualizar reportes.</p>  | <p>El sistema muestra una lista con los tipos de reportes que se pueden visualizar, muestra los campos correspondientes para filtrar el reporte por estado y por un rango de fechas.</p>  | <p>El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas.</p> |
| <p>EC 2.1: El actor selecciona el tipo de reporte “Denuncias” para visualizarlo, sin aplicar</p>   | <p>El sistema busca en la base de datos la cantidad existente de “Denuncias” pertenecientes a los estados de Venezuela, si es</p>   | <p>El actor debe estar autenticado y las capas bases del Mapa deben estar cargadas.</p> |

## CAPÍTULO 4

|   |   |  |
|---|---|--|
| ningún filtro.  | cantidad es igual a 0, el sistema muestra un mensaje de alerta con el siguiente texto “No se han registrado Denuncias en el sistema”. |  |
| EC 2.2: El actor selecciona el tipo de reporte “Denuncias” que desea visualizar en el mapa, selecciona filtrarlo por un rango de fecha:<br><br>Fecha inicial: 10/01/2014<br><br>Fecha final: 21/05/2013 | El sistema muestra un mensaje de alerta con el siguiente texto “El rango de fecha establecido es incorrecto”.                         |  |

Tabla 7. Pruebas de Caja Negra del CU Visualizar Reporte.

### 4.3 Resultados de las Pruebas aplicadas al módulo Mapificación

A continuación se muestran en la Tabla 8 y en la gráfica de la Figura 17 los resultados de las pruebas de caja negra que fueron realizadas al módulo Mapificación en las que se puede observar el número de iteraciones que se le aplicaron los casos de pruebas al módulo, el total de no conformidades<sup>8</sup> encontradas y las que fueron resueltas, e igualmente se muestra la cantidad que no procedieron.

#### 4.3.1 Resultados de las pruebas de caja negra aplicadas al módulo Mapificación

| No. de Iteraciones | Total de no conformidades | No Procede | Resueltas |
|--------------------|---------------------------|------------|-----------|
| 1ra Iteración      | 7                         | 1          | 6         |
| 2da Iteración      | 2                         | 0          | 2         |
| 3ra Iteración      | 0                         | 0          | 0         |

Tabla 8. Resultados de las pruebas de caja negra aplicadas al módulo Mapificación.

<sup>8</sup> no conformidades: problemas detectados al probar la aplicación.

## CAPÍTULO 4

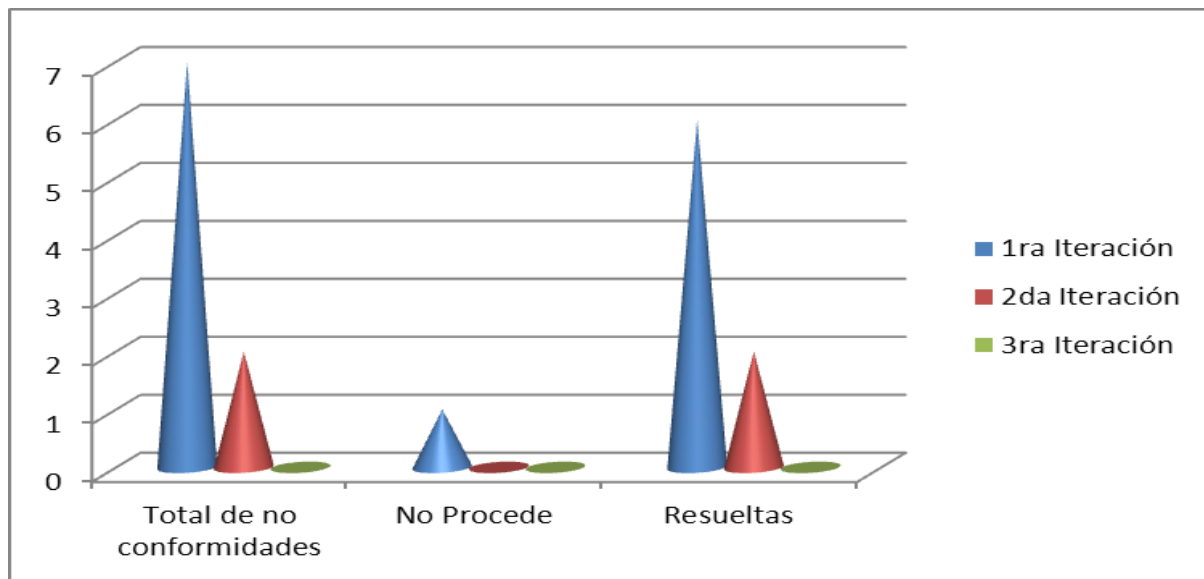


Figura 17. Resultados de las pruebas de caja negra aplicadas al módulo Mapificación.

#### 4.4 Conclusiones

La elaboración y ejecución de casos de prueba de caja blanca y de caja negra permitió identificar y solucionar de las deficiencias detectadas en el módulo Mapificación. Las pruebas de caja blanca se aplicaron sobre el código fuente del módulo, con el propósito de encontrar errores en las consultas que se realizan a la base de datos. Las pruebas de caja negra posibilitaron verificar las funcionalidades que debe cumplir el módulo.

## CONCLUSIONES

### CONCLUSIONES

Con la culminación del trabajo de diploma Diseño e Implementación del módulo Mapificación para el Sistema de Prevención del Delito de Venezuela, se dio cumplimiento a los objetivos propuestos.

El estudio realizado a los SIG desarrollados en el mundo, en Cuba y en la UCI permitió seleccionar las herramientas y tecnologías que más se ajustan a la situación actual del Sistema de Prevención del Delito de Venezuela. Además se adoptaron las ventajas y desventajas de estas aplicaciones informáticas, para que sirvieran de base de conocimientos en la implementación del módulo.

Se propuso una solución, la cual posibilitó tener una concepción más amplia de lo que se debía desarrollar y se especificaron las funcionalidades que tenía que cumplir el módulo Mapificación.

Con el diseño e implementación de la solución propuesta se le dio cumplimiento a los requisitos que debía cumplir el módulo Mapificación.

La realización de un conjunto de pruebas al módulo Mapificación logró determinar que el mismo cumple con todas las funcionalidades especificadas.

## RECOMENDACIONES

### RECOMENDACIONES

Escalar el módulo Mapificación para que sea capaz de representar sobre mapas digitales la información asociada a nuevas acciones que se añadan en el Sistema de Prevención del Delito de Venezuela con el objetivo de contribuir al proceso de toma de decisiones.

## BIBLIOGRAFÍA

## BIBLIOGRAFÍA

1. **GONZÁLEZ DOMECH, Armando.** Observatorio Geográfico de América Latina. *La Cartografía de Atlas en Cuba: Situación actual y perspectivas en el siglo XXI.* [En línea] [Citado el: 15 de enero de 2013.]  
<http://www.observatoriogeograficoamericalatina.org.mx/egal9/Nuevatecnologias/Sig/01.pdf>. ISBN: 959-7160-12-9.
2. **CLARKE, Keith.** *Analytical and Computer Cartography.* New Jersey : Prentice Hall Professional Technical Reference, 1990. ISBN: 0130334812 ISBN-13: 9780130334817.
3. **BRINEY, Amanda.** Geography. *An Overview of Geographic Information Systems.* [En línea] 6 de Abril de 2008. [Citado el: 2 de febrero de 2013.]  
<http://geography.about.com/od/geographyintern/a/gisoverview.htm>
4. **PERVERSI, Ignacio.** Ingeniería Industrial. *Aplicación de Minería de Datos para la exploración y detección de Patrones Delictivos en Argentina.* Buenos Aires : s.n., 2007.
5. **IZNAGAR DE NISTAL, María; BORTHAGARAY, Juan Manuel.** *Mapa del Delito.* Buenos Aires : Nobuko, 2007. ISBN: 9789875841307.
6. **Mapas Digitales S.A** Dmapas. *Seguridad Ciudadana y Gestión Policial (STEGPOL).* [En línea] [Citado el: 9 de enero de 2013.]  
[http://www.dmapas.com/productos\\_stegpol.htm](http://www.dmapas.com/productos_stegpol.htm).
7. **Sitio Web de la empresa Mapalia Networks.** Mapalia Networks. *Mapa de la Delincuencia.* [En línea] Mapalia Network, 15 de diciembre de 2008. [Citado el: 12 de diciembre de 2012.] <http://www.mapalia/c/delitos>.
8. **Sitio Web Oficial de Citizens Report.** Citizens Report. *Citizens leading on safer streets and homes.* [En línea] ShakaZoom, abril de 2011. [Citado el: 9 de enero de 2013.] <http://www.citizensreportuk.org/>.
9. **DEMPSEY, Caitlin.** GisLonge. *Crime Mapping and the Los Angeles Police.* [En línea] 13 de Mayo de 2011. [www.gislounge.com](http://www.gislounge.com).



## BIBLIOGRAFÍA

10. **POTTS, Alejandro; VILLEGAS, Lianne.** Universidad de las Ciencias Informáticas. *Modernización de los Sistemas del Centro de Información y Mando de la Unidad Provincial de Patrullas.* La Habana : s.n., 2010.
11. **SALDÍVAR, Yoenis.** Universidad de las Ciencias Informáticas. *Modelo de desarrollo basado en líneas de productos de software para Sistemas de Información Geográfica sobre la base de la Plataforma GeneSIG.* La Habana : s.n., 2012.
12. **KRUCHTEN, Philippe.** *The Rational Unified Process: An Introduction.* Boston - Estados Unidos : Addison-Wesley, 2000. ISBN: 0201707101.
13. **JACOBSON, Ivar, et al.** *El Proceso Unificado de Desarrollo de Software.* s.l. : ADDISON WESLEY, 2007. ISBN-10: 8478290362 ISBN-13: 978-8478290369.
14. **Sitio Web Oficial de GeoServer.** GeoServer. *GeoServer.* [En línea] GeoServer. [Citado el: 13 de diciembre de 2012.] <http://geoserver.org>.
15. **Sitio Web Oficial de OpenLayers.** OpenLayers. *OpenLayers.* [En línea] MetaCarta. [Citado el: 12 de diciembre de 2012.] <http://openlayers.org/>.
16. **HAZZARD, Erik.** *OpenLayers 2.10, Beginner's Guide.* Birmingham, UK : Packt Publishing Ltd, 2011. ISBN 978-1-849514-12-5.
17. **Sitio Web Oficial de PostgreSQL.** PostgreSQL. *PostgreSQL.* [En línea] The PostgreSQL Global Development Group , 1996-2013. [Citado el: 9 de enero de 2012.] <http://www.postgresql.org>.
18. **OBE, Regina; HSU, Leo.** *PostGIS in Action.* Stamford : Manning Publications Co., 2011. ISBN: 978-1-935182269.
19. **Sitio Web Oficial de Oracle.** Oracle Corporation. *Oracle.* [En línea] Oracle Corporation. [Citado el: 10 de febrero de 2013.] <http://www.oracle.com/technetwork/java/javaee/overview/index.html>.
20. **Sitio Web Oficial de Apache Tomcat.** Apache Tomcat. *Apache Tomcat.* [En línea] Apache Software Foundation. [Citado el: 10 de diciembre de 2012.] <http://www.apachetomcat.org>.
21. **Diseño web y desarrollo web.** *Programación en Castellano.* [En Línea] 1998-2013. [http://www.programacion.com/articulo/eclipse -- i - historia\\_y\\_toma\\_de\\_contacto\\_288](http://www.programacion.com/articulo/eclipse_-_i_-_historia_y_toma_de_contacto_288)

## BIBLIOGRAFÍA

22. **MARAÑÓN, Gonzalo.** Instituto de Física Aplicada del CSIC. *Características del lenguaje Java*. [En línea] CSIC, 1999. [Citado el: 12 de diciembre de 2012.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
23. **GOODMAN, Danny.** *JavaScript Bible*. New York : Hungry Minds, Inc, 2001. ISBN:0764547186.
24. **MORALES, Mav.** Universidad de Chile. *Framework*. [En línea] 2007. [Citado el: 15 de enero de 2013.] [http://www.tesis.uchile.cl/tesis/uchile/2007/vasquez\\_mm/sources/vasquez\\_mm.pdf](http://www.tesis.uchile.cl/tesis/uchile/2007/vasquez_mm/sources/vasquez_mm.pdf).
25. **Sitio Web Oficial de Spring Framework.** Spring Framework. [En línea] 2013. [Citado el: 10 de enero de 2013.] <http://www.springframework.org>.
26. **BENAVIDES, Yadira; GÓMEZ, Juan Carlos.** Diseño e implementación de los módulos Decisiones y Egresos del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela. Ciudad de La Habana : s.n., 2008.
27. **Sitio Web Oficial de Lenguaje Unificado de Modelado.** *Lenguaje Unificado de Modelado (UML)*. [En línea] [Citado el: 12 de enero de 2013.] <http://www.uml.org>.
28. **SOMMERVILLE, Ian.** *Ingeniería de Software. 6ta. Edición*. Nueva Jersey : Prentice Hall, 2002. ISBN:970-26-0206-8.
29. **TEDESCHI, Nicolás.** MSDN Microsoft. *¿Qué es un Patrón de Diseño?*. [En línea] [Citado el: 13 de febrero de 2013.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx#XSLTsection123121120120>.
30. **WELICKI, León.** MSDN. *Patrones de Fabricación: Fábricas de Objetos*. [En línea] Microsoft. [Citado el: 7 de febrero de 2013.] <http://msdn.microsoft.com/es-es/library/bb972258.aspx#authorbrief>.
31. **WELICKI, León.** MSDN. *El Patrón Singleton*. [En línea] [Citado el: 7 de febrero de 2013.] <http://msdn.microsoft.com/es-es/library/bb972272.aspx>.
32. **OBE, Regina; HSU, Leo.** *PostGIS in Action*. Stamford : Manning Publications Co., 2011. ISBN: 978-1-935182269.
33. **Sitio Web Oficial de Hibernate.** Hibernate. *Hibernate Framework*. [En línea] Hibernate. [Citado el: 12 de enero de 2013.] <http://www.hibernate.org>.

## BIBLIOGRAFÍA

34. **Apache Software Foundation.** *Apache Tomcat.* [En línea] APACHE. [Citado el: 9 de enero de 2013.] <http://apachefoundation.wikispaces.com/Apache+Tomcat>.
35. **VOZMEDIANO, Laura; CÉSAR, San Juan .***Empleo de Sistemas de Información Geografica en el estudio del Miedo al Delito..* 4, Madrid : Revista Española de Investigación Criminológica, 2006. ISSN: 1696-9219.
36. **KROPLA, Bill.** *Beginning Mapserver: Open Souerce GIS Development.* New York : Springer, 2005. ISBN (pbk):1-59059-490-8.
37. **LÓPEZ, Inglis.** Sitio Oficial de la Universidad de Granada. *Análisis Espacial de Datos Mediante Sistemas de Información Geográfica: Propuesta Metodológica.* [En línea] <http://www.ugr.es/~cmetodo/pdf/comunicaciones/inglis.pdf>.

## GLOSARIO DE TÉRMINOS

### GLOSARIO DE TÉRMINOS

**ArcSDE:** Es una tecnología que pertenece al ESRI que sirve para acceder y administrar datos geoespaciales dentro de las bases de datos relacionales.

**CGI:** Es una tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

**DB2:** Es un motor de base de datos relacional que integra XML de manera nativa, permite almacenar documentos completos dentro del tipo de datos XML.

**ECW:** Es formato de archivo propietario para imágenes ráster.

**ESRI (Environmental Systems Research Institute):** Es una empresa que desarrolla y comercializa software para Sistemas de Información Geográfica, es una de las compañías líderes en el sector a nivel mundial.

**GeoTIFF:** Es un estándar de metadatos de dominio público que permite que la información geo-referenciada sea encajada en un archivo de imagen de formato TIFF.

**GIF (CompuServe GIF):** Es un formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.

**GML (Geography Markup Language):** Es un sub-lenguaje de XML descrito como una gramática en XML para el modelaje, transporte y almacenamiento de información geográfica.

**GNU LGPL:** Es una licencia de software que pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios.

**GTOPO30 (Global Topographic Data):** Es un modelo de elevación digital global con una resolución de 30 segundos de arco (aproximadamente 1km).

**Image Mosaics:** Es un almacén de datos permite la creación de mosaicos o teselas a partir de un número de tramas geo-referenciadas.

**JPEG (Joint Photographic Experts Group):** Es un estándar de compresión y codificación de archivos de imágenes, es considerado como un formato de archivo.

## GLOSARIO DE TÉRMINOS

**JPEG2000:** Es un estándar de compresión y codificación digital de imágenes, trabaja con niveles de compresión mayores que los de JPEG.

**JUnit:** Es un conjunto de bibliotecas que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

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

**MapBasis:** Es el ambiente de desarrollo para el SIG MapInfo.

**MapInfo:** Es una herramienta de Sistemas de Información Geográfica que le permite realizar diversos y complejos análisis geográficos ideales para facilitar la toma de decisiones.

**MapServer:** Es una plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas interactivas para la web.

**MrSID:** Es un estándar abierto de compresión de imágenes ráster.

**MySQL:** Es un sistema gestor de bases de datos relacional.

**OCG (Open Geospatial Consortium):** Consorcio que agrupa más de 370 organizaciones públicas y privadas, su objetivo es definir estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica y de la World Wide Web.

**Oracle Spatial:** Es una opción que ofrece Oracle Database para gestionar toda la información geoespacial, incluidos datos vectoriales y ráster es totalmente compatible con servicios web.

**PDF (Portable Document Format):** Es un formato de almacenamiento de documentos digitales.

**PNG (Portable Network Graphics):** Es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes.

**Shapefile:** Es un formato vectorial de almacenamiento digital donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos.

**SLD:** Es un archivo que va a establecer el estilo con el cual se van a mostrar las capas en el mapa.

## GLOSARIO DE TÉRMINOS

**SVG (Scalable Vector Graphics):** Es una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados.

**WCS (Web Coverage Service):** Es un servicio que proporciona una interfaz que permite realizar peticiones de cobertura geográfica a través de la web, cumple con los estándares definidos por el OCG.

**Web Geoespacial:** Es la combinación de información geográfica con la información abstracta que predomina en la web, generando contextos que permiten realizar búsquedas u ofertar servicios en base a la localización.

**WFS (Web Feature Service):** Es un servicio orientado a facilitar el intercambio de información geográfica a través de Internet, cumple con los estándares definidos por el OCG.

**WMS (Web Map Service):** Es un servicio que permite crear mapas de datos referenciados de forma dinámica a partir de información geográfica, cumple con los estándares definidos por el OCG.

**XML (eXtensible Markup Language):** Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C).

## ANEXOS

### Anexo 1 .Clases Entidades del Dominio

#### Anexo 1.1 Clase Entidad Estado

|  |
|--|
| <b>Estado</b>                            |
| -Integer id;                             |
| -Boolean activo;                         |
| -String descripcion;                     |
| -String nombre;                          |
| -Boolean permanente;                     |
| +Estado()                                |
| +Integer getId()                         |
| +void setId(Integer id)                  |
| +Boolean getActivo()                     |
| +void setActivo(Boolean activo)          |
| +String getDescripcion()                 |
| +void setDescripcion(String descripcion) |
| +String getNombre()                      |
| +void setNombre(String nombre)           |
| +Boolean getPermanente()                 |
| +void setPermanente(Boolean permanente)  |
| +int hashCode()                          |
| +boolean equals(Object obj)              |

#### Anexo 1.2 Clase Entidad Municipio

|                  |
|------------------|
| <b>Municipio</b> |
| -MunicipioPK id; |
| -Boolean activo; |

## ANEXOS

|   |
|---|
| -String descripcion;                      |
| -String nombre;                           |
| -Boolean permanente;                      |
| -Estado estado;                           |
| -Integer idMunicipio                      |
| -Integer idEstado;                        |
| +Integer getIdMunicipio()                 |
| +void setIdMunicipio(Integer idMunicipio) |
| +Integer getIdEstado()                    |
| +void setIdEstado(Integer idEstado)       |
| +Municipio()                              |
| +MunicipioPK getId()                      |
| +setId(MunicipioPK id)                    |
| +Boolean getActivo()                      |
| +void setActivo(Boolean activo)           |
| +String getDescripcion()                  |
| +void setDescripcion(String descripcion)  |
| +String getNombre()                       |
| +void setNombre(String nombre)            |
| +Boolean getPermanente()                  |
| +setPermanente(Boolean permanente)        |
| +Estado getEstado()                       |
| +void setEstado(Estado estado)            |
| +int hashCode()                           |
| +boolean equals(Object obj)               |



## ANEXOS

## Anexo 1.3 Clase Entidad Denuncia

|   |
|---|
| <b>Denuncia</b>                               |
| Long id;                                      |
| TipoActividad tipoActividad;                  |
| String datos;                                 |
| String descripcionDenunciado;                 |
| String direccion;                             |
| Date fecha;                                   |
| Time hora;                                    |
| Lugar lugar;                                  |
| String otroLugar;                             |
| String otroOrigen;                            |
| String referido;                              |
| EstatusDenuncia estatus;                      |
| Ciudadano ciudadano;                          |
| UsuarioSistema usuario;                       |
| ClasificacionDelito clasificacionDelito;      |
| OrigenDenuncia origen;                        |
| Sector sector;                                |
| TipoDenuncia tipo;                            |
| List<Remision> listaRemision;                 |
| CoordinacionRegional coordinacion;            |
| Denuncia()                                    |
| Long getId()                                  |
| void setId(Long id)                           |
| TipoActividad getTipoActividad()              |
| setTipoActividad(TipoActividad tipoActividad) |

## ANEXOS

|  |
|--|
| String getDatos()  |
| void setDatos(String datosDenuncia)                                  |
| String getDescripcionDenunciado()                                    |
| void setDescripcionDenunciado(String descripcionDenunciado)          |
| String getDireccion()  |
| void setDireccion(String direccion)                                  |
| Date getFecha()  |
| void setFecha(Date fecha)  |
| Lugar getLugar()   |
| void setLugar(Lugar lugar)   |
| String getOtroLugar()  |
| void setOtroLugar(String otroLugar)                                  |
| String getOtroOrigen()   |
| setOtroOrigen(String otroOrigen)                                     |
| String getReferido()   |
| void setReferido(String referido)                                    |
| EstatusDenuncia getEstatus()   |
| void setEstatus(EstatusDenuncia status)                              |
| Ciudadano getCiudadano()   |
| void setCiudadano(Ciudadano ciudadano)                               |
| ClasificacionDelito getClassificacionDelito()                        |
| void setClasificacionDelito(ClasificacionDelito clasificacionDelito) |
| OrigenDenuncia getOrigen()   |
| setOrigen(OrigenDenuncia origen)                                     |
| Sector getSector()   |
| void setSector(Sector sector)  |
| void setUsuario(UsuarioSistema usuario)                              |

## ANEXOS

|   |
|---|
| UsuarioSistema getUsuario()                             |
| void setTipo(TipoDenuncia tipo)                         |
| TipoDenuncia getTipo()                                  |
| void setListaRemision(List<Remision> listaRemision)     |
| List<Remision> getListaRemision()                       |
| void setHora(Time hora)                                 |
| Time getHora()  |
| void setCoordinacion(CoordinacionRegional coordinacion) |
| oordinacionRegional getCoordinacion()                   |
| int hashCode()  |
| boolean equals(Object obj)                              |

**Anexo 1.4 Clase Entidad Actividad**

|   |
|---|
| <b>Actividad</b>  |
| Long id;  |
| Date fecha;   |
| Date fechaFin;  |
| Time horaFin;   |
| Time horalnicio;  |
| String lugar;   |
| String nombre;  |
| String caractPoblacion;                                       |
| Parroquia parroquia;  |
| RProyectoActividadTipo rproyectoActividadTipo;                |
| CoordinacionRegional coordinacionRegional;                    |
| List<ActividadInstitucionOrganizacion> listadoActTipolnstOrg; |
| List<RedesArticulacion> listadoRedesArticulacion;             |
| Actividad()   |

## ANEXOS

|   |
|---|
| Actividad(Date fecha, Date fechaFin, Time horaFin, Time horalnicio,<br>String lugar, String nombre, String caractPoblacion,<br>Parroquia parroquia, RProyectoActividadTipo rproyectoActividadTipo)) |
| Long getId()  |
| void setId(Long id)   |
| String getCaractPoblacion()   |
| void setCaractPoblacion(String caractPoblacion)   |
| Date getFecha()   |
| void setFecha(Date fecha)   |
| Date getFechaFin()  |
| void setFechaFin(Date fechaFin)   |
| Time getHoraFin()   |
| void setHoraFin(Time horaFin)   |
| Time getHoralnicio()  |
| void setHoralnicio(Time horalnicio)   |
| String getLugar()   |
| void setLugar(String lugar)   |
| String getNombre()  |
| void setNombre(String nombre)   |
| CoordinacionRegional getCoordinacionRegional()  |
| void setCoordinacionRegional(CoordinacionRegional coordinacionRegional)   |
| RProyectoActividadTipo getRproyectoActividadTipo()  |
| void setRproyectoActividadTipo(RProyectoActividadTipo rproyectoActividadTipo)   |
| Parroquia getParroquia()  |
| setParroquia(Parroquia parroquia)   |
| List<ActividadInstitucionOrganizacion> getListadoActTipolnstOrg()   |
| void setListadoActTipolnstOrg(List<ActividadInstitucionOrganizacion> listadoActTipolnstOrg)   |

## ANEXOS

|  |
|--|
| List<RedesArticulacion> getListadoRedesArticulacion()                              |
| void setListadoRedesArticulacion(List<RedesArticulacion> listadoRedesArticulacion) |
| int hashCode()   |

**Anexo 1.5 Clase Entidad ActividadPlanificada**

|   |
|---|
| ActividadPlanificada                        |
| String justificacion;                       |
| Status status;                              |
| ActividadPlanificada()                      |
| String getJustificacion()                   |
| void setJustificacion(String justificacion) |
| Status getStatus()                          |
| void setStatus(Status status)               |

**Anexo 1.6 Clase Entidad ActividadRealizada**

|   |
|---|
| ActividadRealizada  |
| -String fortalezas;   |
| -String debilidades;  |
| -Accesibilidad accesibilidad;   |
| -Float costoGob;  |
| -Float costoDgpd;   |
| -Long idActividadPlanificada;   |
| -List<IndicadorVersion> listadoIndicadorVersion;                              |
| -List<ActividadRealizadaRecursos> listadoActRealRecursos;                     |
| -List<ActividadRealizadaMediosComunicacion> listadoActRealMediosComunicacion; |
| -List<ActividadRealizadaEncuestaComunidad> listadoActRealEvaluacion;          |
| +ActividadRealizada()   |

## ANEXOS

|  |
|--|
| +ActividadRealizada(Date fecha, Date fechaFin, Time horaFin,<br>Time horahicio, String lugar, String nombre,<br>String caractPoblacion, Parroquia parroquia,<br>RProyectoActividadTipo rproyectoActividadTipo<br>) |
| +String getFortalezas()  |
| +void setFortalezas(String fortalezas)   |
| +String getDebilidades()   |
| +void setDebilidades(String debilidades)   |
| +Accesibilidad getAccesibilidad()  |
| +void setAccesibilidad(Accesibilidad accesibilidad)  |
| +Float getCostoGob()   |
| +void setCostoGob(Float costoGob)  |
| +Float getCostoDgpd()  |
| +void setCostoDgpd(Float costoDgpd)  |
| +Long getIdActividadPlanificada()  |
| +void setIdActividadPlanificada(Long idActividadPlanificada)   |
| +List<IndicadorVersion> getListadoIndicadorVersion()   |
| +void setListadoIndicadorVersion(List<IndicadorVersion> listadoIndicadorVersion)   |
| +List<ActividadRealizadaRecursos> getListadoActRealRecursos()  |
| +void setListadoActRealRecursos(List<ActividadRealizadaRecursos><br>listadoActRealRecursos)  |
| +List<ActividadRealizadaMediosComunicacion> getListadoActRealMediosComunicacion()  |
| +void setListadoActRealMediosComunicacion(<br>List<ActividadRealizadaMediosComunicacion> listadoActRealMediosComunicacion)   |
| +List<ActividadRealizadaEncuestaComunidad> getListadoActRealEvaluacion()   |
| +void setListadoActRealEvaluacion(List<ActividadRealizadaEncuestaComunidad><br>listadoActRealEvaluacion)   |

## ANEXOS