

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2



*Integración del Sistema de Mapificación Web y
el Sistema de Gestión Policial*

Trabajo de Diploma

Presentado para optar por el título de Ingeniero en Ciencias Informáticas.

Autores:

- ✓ Mayte Mozo Reyes
- ✓ Carlos Rafael Labrada Arce

Tutor:

Ing. Reynier González Tejeda

“Año 52 de la Revolución”

24 de Junio de 2010, Ciudad de la Habana, Cuba.

Declaración de Autoría

Manifestamos ser autores de la tesis en cuestión y conferimos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Como constancia se firma la presente a los ____ días del mes _____ del año _____.

Mayte Mozo Reyes

Autor

Carlos Rafael Labrada Arce

Autor

Ing. Reynier González Tejeda

Tutor

Dedicatoria

Mayte:

Le dedico este trabajo en primer lugar a mi mamá, quien me ha dado fuerzas para seguir adelante a lo largo de estos cinco años, al bichito a quien quiero mucho, a mis abuelas aunque no están físicamente, A mi pareja por su incondicional apoyo y comprensión, a mi papá por su guía profesional a lo largo de mis estudios, a toda mi familia y amigos.

Carlos:

A mis padres por todo el amor y cariño que me han dado en todos estos años, mis hermanos por la dedicación y preocupación durante mi vida estudiantil. A mi familia que los quiero mucho, a mi novia y los amigos que siempre han estado a mi lado.

Agradecimientos

Le agradecemos a nuestro tutor Rey por su empeño para que el trabajo fuera lo mejor posible. A Franky por su ayuda incondicional, gracias a el abusador por su gran apoyo en los momentos de cólera, al Tigre, al Figura, al equipo de SIGEPOL en general y a Potter. Agradecemos también al arquitecto del proyecto SIGMCI Antonio Membrides Espinosa por su magnífico asesoramiento. A la profesora Mayda Gil muchas gracias. A todas aquellas personas que de una forma u otra contribuyeron con la realización de este trabajo.

A la Revolución Cubana.

Resumen

Para el control de las labores internas de seguridad de la Policía Nacional Bolivariana de Venezuela fue desarrollado el Sistema de Gestión Policial (SIGEPOL); aplicación encargada de tramitar considerable cantidad de información acerca de las dependencias policiales. Estas entidades tienen como misión velar por la seguridad de sus ciudadanos, mediante actividades policiales como el control de los hechos delictivos, así como de la ubicación geográfica de cada uno de ellos.

SIGEPOL manipula suficiente información geográfica la cual demanda ser representada en un mapa para lograr un eficiente monitoreo y control del funcionamiento y localización de las acciones que se llevan a cabo sobre un área determinada en un momento dado.

Información geográfica se define como la descripción de objetos, lugares, eventos referenciados o lugares relativos a la superficie terrestre, que poseen coordenadas o regiones definidas explícitamente, de manera que pueden ser colocados en un mapa.

Con el fin de representar y procesar información geográficamente referenciada, para uso exclusivo de las dependencias policiales, y con el objetivo de mejorar los servicios de procesamiento de la información geográfica que se maneja, fue desarrollado el Sistema de Mapificación Web (SMW) el cual incluye operaciones básicas de navegación para el trabajo con mapas.

Con el objetivo de mejorar la eficiencia y eficacia de las prestaciones que ofrece SIGEPOL, permitiendo el buen desenvolvimiento de los funcionarios que utilicen el sistema para el cumplimiento de sus actividades, nace la idea de perfeccionar el SMW de manera que permita visualizar y procesar la información geográfica que maneja SIGEPOL.

Con el desarrollo de la solución propuesta, los usuarios finales del SMW tendrán la posibilidad de realizar con apoyo de los mapas, análisis más profundos y estudios estadísticos policiales que contribuirán en la toma de decisiones y estrategias a seguir ante situaciones determinadas. Los mapas ofrecen a los analistas representaciones geográficas, pues el mapeo del delito ayuda a los funcionarios de las dependencias policiales a proteger, de manera más efectiva, a los ciudadanos en sus respectivas áreas de responsabilidad.

Índice

Introducción	I
Capítulo 1. Fundamentación Teórica	1
1.1 Introducción.....	1
1.2 Seguridad Ciudadana en Venezuela	1
1.2.1 Policía Nacional Bolivariana.....	1
1.3 Sistema de Gestión Policial	3
1.3.1 Módulo de Reseña	4
1.3.2 Módulo de Denuncia.....	4
1.3.3 Módulo de Operativos Policiales.....	4
1.3.4 Módulo de Dependencia	5
1.4 Sistema de Información Geográfica (SIG)	5
1.4.1 Aplicaciones de los SIG	5
1.4.2 Sistema de Mapificación Web.....	7
1.5 Tecnología SIG a utilizar.....	8
1.5.1 MapInfo MapXtreme Java Edition	9
1.6 Plataforma de Software a utilizar.....	9
1.6.1 La Plataforma Java.....	9
1.7 Frameworks a utilizar.....	10
1.7.1 Ext JS.....	10
1.7.2 Spring Framework	11
1.7.3 Ibatis	13
1.8 Lenguaje Unificado de Modelado (UML)	13
1.9 Metodología de Desarrollo de Software	13
1.9.1 Metodología Proceso Unificado de Desarrollo (RUP)	14
1.10 Herramientas de Desarrollo	16
1.10.1 Eclipse	16
1.10.2 Herramienta CASE	17
1.10.2.1 Visual Paradigm	17
1.12 Conclusiones.....	17
Capítulo 2. Características del Sistema	19
2.1 Introducción.....	19

2.2 Propuesta del Sistema.....	19
2.2.1 Descripción General.....	19
2.3 Modelo del Dominio	20
2.3.1 Conceptos del Modelo de Dominio	20
2.3.2 Descripción del Modelo de Dominio	21
2.4 Requerimientos	21
2.4.1 Definición de Requerimientos Funcionales	22
2.4.2 Definición de Requerimientos No Funcionales	25
2.5 Modelo del Sistema.....	29
2.5.1 Actores del Sistema.....	29
2.5.2 Paquetes	29
2.5.3 Diagramas de Casos de Uso del Sistema	30
2.5.4 Descripción de los Casos de Uso	31
2.6 Conclusiones.....	35
Capítulo 3. Análisis y Diseño del Sistema	37
3.1 Introducción.....	37
3.2 Análisis	37
3.2.1 Diagrama de Clases del Análisis Paquete Operaciones Básicas	38
3.3 Arquitectura	39
3.3.1 Propuesta de Arquitectura.....	39
3.4 Diseño	41
3.4.1 Diagramas de Interacción	42
3.4.2 Diagrama de Clases del Diseño.....	42
3.4.3 Diagrama de Paquetes del Diseño	43
3.4.4 Diagrama de Clases del Diseño Paquete Núcleo	45
3.5 Modelo Lógico de Datos	45
3.6 Modelo Físico de datos	47
3.7 Conclusiones.....	48
Capítulo 4. Implementación	49
4.1 Introducción.....	49
4.2 Modelo de Implementación.....	49
4.2.1 Diagrama de Despliegue	49

4.2.2 Diagrama de Componentes	50
4.3 Conclusiones.....	56
Capítulo 5. Estudio de Factibilidad.....	57
5.1 Introducción.....	57
5.2 Estimación de Puntos por Caso de Uso.....	57
5.2.1 Cálculo de Puntos de Caso de Usos sin Ajustar	57
5.2.5 Costo del Proyecto	61
5.2.6 Cálculo del Tiempo de Desarrollo del Proyecto.....	62
5.3 Beneficios Tangibles e Intangibles	62
5.4 Conclusiones.....	62
Conclusiones Generales.....	63
Recomendaciones.....	64
Referencias Bibliográficas.....	65
Bibliografía	67
Glosario de Términos.....	68

Introducción

En la República Bolivariana de Venezuela debido al alto índice delictivo que existe, afianzado por la oposición al gobierno de Hugo Chávez la Policía Nacional Bolivariana se traza como objetivo la automatización de las actividades policiales haciendo uso de las TIC¹, con el fin de proveer una plataforma tecnológica de avanzada que permita el eficiente cumplimiento de sus tareas.

Por esta razón el MPPRIJ² promueve el desarrollo de SIGEPOL para el uso exclusivo de la Policía Nacional Bolivariana en sus labores internas de seguridad. Esta aplicación se encarga de tramitar considerable cantidad de información acerca de las dependencias policiales. Estas entidades tienen como misión velar por la seguridad de sus ciudadanos, mediante actividades policiales como el control de los hechos delictivos, así como de la ubicación geográfica de cada uno de ellos.

SIGEPOL manipula suficiente información geográfica la cual exige ser representada en un mapa para lograr un eficiente monitoreo y control del funcionamiento y localización de las acciones que se llevan a cabo en un momento determinado.

Se conoce como información geográfica a aquella que describe lugares, objetos o eventos referenciados sobre la superficie terrestre, que tienen coordenadas o regiones definidas explícitamente, de forma que pueden ser ubicados en un mapa. (1)

Con el fin de representar y procesar información geográficamente referenciada, para uso exclusivo de las dependencias policiales de la República Bolivariana de Venezuela fue desarrollado el SMW. Este sistema brinda funcionalidades que facilitan la realización de estudios estadísticos policiales, además de poseer las operaciones básicas para la navegación sobre mapas. Sin embargo el SMW no brinda la posibilidad de visualizar la información geográfica que es manejada por SIGEPOL para un mejor desempeño de los funcionarios en el cumplimiento de sus tareas, facilitando la identificación de zonas de alta criminalidad y permitiendo así concentrar los esfuerzos hacia esas áreas.

¹ Tecnologías de la Información y las Comunicaciones.

² Ministerio del Poder Popular para Relaciones Interiores y de Justicia.

Ante la **situación problémica** anteriormente expuesta, se plantea como **problema científico** que el SMW carece de funcionalidades que permitan visualizar la información geográficamente referenciada existente en SIGEPOL.

La investigación tiene como **objeto de estudio** las aplicaciones que visualizan información geográfica estableciendo como **campo de acción** los sistemas web de información geográfica como soporte a los órganos policiales.

Se enuncia como **objetivo general** realizar la integración de SIGEPOL y SMW, mediante el desarrollo de nuevas funcionalidades sobre este último, que permita una mejor apreciación de la información geográfica presente en SIGEPOL.

Del objetivo general trazado, surgen los siguientes **objetivos específicos**:

- ✓ Analizar trabajos precedentes que presentan como argumento la visualización de la información geográfica mediante sistemas web como soporte a los órganos policiales.
- ✓ Especificar las funcionalidades a desarrollar en el SMW para dar soporte al procesamiento de información geográfica que presenta SIGEPOL.
- ✓ Rediseñar el modelo de la arquitectura del SMW.
- ✓ Obtener una versión funcional del SMW que posibilite la representación visual de la información geográficamente referenciada de SIGEPOL.

Para dar cumplimiento a los objetivos específicos de la investigación planteados, se proponen las siguientes **tareas de investigación**:

- ✓ Estudiar las herramientas, frameworks y la metodología a emplear en el desarrollo de la solución propuesta.
- ✓ Estudiar los procesos de negocio de SIGEPOL para identificar la información geográficamente referenciada presente en el mismo.
- ✓ Modificar la arquitectura del SMW de manera que se ajuste a las nuevas funcionalidades.
- ✓ Implementar las funcionalidades para el procesamiento de información geográfica de SIGEPOL sobre el SMW, para validar la solución propuesta.

Con el fin de obtener una solución que satisfaga el problema explicado anteriormente y dar cumplimiento al objetivo general, es de vital importancia comprender los procesos de SIGEPOL para la correcta identificación de la información geográfica a representar de este sistema. El producto final de esta investigación es la integración de SIGEPOL y el SMW mediante la creación de nuevas funcionalidades sobre este último, que permitan visualizar la información geográficamente referenciada de SIGEPOL.

El documento se compone por cinco capítulos:

Capítulo 1: **Fundamentación Teórica** se presentan los principios que se utilizan como sustento teórico para la solución del problema, así como se exponen las herramientas, el lenguaje de programación, además de la metodología de desarrollo de software a utilizar.

Capítulo 2: **Características del Sistema** se describe el modelo del dominio del sistema donde se presentan los principales conceptos ajustados a las nuevas funcionalidades a implementar en el SMW. Se detallan los requerimientos funcionales como características que el sistema debe cumplir, donde se especifica con qué información geográfica de SIGEPOL se trabaja. Entre los requerimientos no funcionales los cuales constituyen cualidades que el producto debe tener se presentan las especificaciones de los mapas a utilizar. También se definen los actores del sistema y se muestran los diagramas de casos de uso del sistema con la descripción correspondiente a cada uno de ellos.

Capítulo 3: **Análisis y Diseño del Sistema** se especifican las clases del análisis identificadas, se muestra la propuesta de arquitectura del sistema, así como los diagramas de clases del diseño por cada paquete definido. También aparece el diagrama de clases persistentes y el diagrama entidad – relación del modelo de datos de la aplicación.

Capítulo 4: **Implementación** se expone el diagrama de despliegue el cual se utiliza para capturar elementos de configuración del procesamiento y las conexiones que existen entre los elementos del sistema; así como permite representar la distribución de los componentes de software en nodos físicos. Se proponen los diagramas de componentes manifestando la transición de los artefactos del diseño en términos de subsistemas de implementación y la relación de dependencia que existe entre ellos.

Capítulo 5: **Estudio de Factibilidad** se realiza una estimación del tiempo y esfuerzo para desarrollar el sistema haciendo uso de la técnica de puntos por casos de uso.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En este apartado se expone de manera precisa la necesidad que posee SMW de brindar funcionalidades que permitan procesar y visualizar información geográficamente referenciada de SIGEPOL, para un mejor desenvolvimiento de las tareas de la policía. También se describe teóricamente la metodología a utilizar, las tecnologías y herramientas de desarrollo necesarias para llevar a cabo la integración de ambos sistemas.

1.2 Seguridad Ciudadana en Venezuela

La seguridad de los residentes constituye un factor imprescindible para el buen funcionamiento de la sociedad de cualquier país a lo largo y ancho del planeta; siendo la misma de primer orden para aquellos gobiernos que se preocupan por este aspecto, constituyendo la hermana República Bolivariana de Venezuela un ejemplo.

A pesar de esto, la nación es sede de una gran actividad delictiva, afianzada por la oposición al gobierno y la tensa situación política que esta genera; con el fin de crear el caos poniendo en peligro la seguridad de los ciudadanos. En torno a esto el MPPRIJ ha tomado cartas en el asunto, trazando estrategias para salvaguardar la coexistencia del país. (2)

Entre los diversos organismos que velan por la seguridad de los ciudadanos en Venezuela se encuentran la Policía Nacional Bolivariana, la Organización de Protección Civil y Administración de Desastres y el CICPC³.

1.2.1 Policía Nacional Bolivariana

La Policía Nacional Bolivariana concibe estrategias para la búsqueda de soluciones pacíficas a los conflictos ciudadanos, caracterizado por sus buenas prácticas, brindando resultados efectivos en términos de la Seguridad Ciudadana y orientados a la consolidación de su proyección nacional e

³ Cuerpo de Investigaciones Científicas, Penales y Criminalísticas.

internacional. Integrados por profesionales de carrera, organizados administrativa y funcionalmente, comprometidos con la participación activa de los ciudadanos en el aseguramiento pacífico, garantías constitucionales y el respeto de los derechos humanos, la aplicación de estrictos controles de gestión y el reconocido uso de tecnología de vanguardia, que contribuya con el buen desempeño de la Policía y el cumplimiento de sus funciones. (3)

Para lograr garantizar la seguridad ciudadana en Venezuela es de gran importancia consolidar las políticas, estrategias, lineamientos y directrices que conlleven a la automatización del control de las acciones policiales, a fin de proveer de tecnología de información y comunicación que permita lograr el cumplimiento de sus tareas sobre una plataforma tecnológica de avanzada.

El MPPRIJ, atendiendo a su misión institucional promueve el desarrollo de SIGEPOL, sistema que tramita la información relacionada con denuncias, reseñas y operativos policiales que se registran en las dependencias de las policías estatales y municipales. Intercambia información con el CICPC y sirve como fuente de información a CTAISC⁴, sincronizándose las maniobras entre estas instituciones, de manera tal que el análisis de los datos compartidos en línea, ayuden a combatir el delito en la República Bolivariana de Venezuela.

Con el objetivo de mostrar datos relevantes sobre acontecimientos y evolución de hechos delictivos en un área geográfica determinada; facilitando el proceso de análisis para estudios estadísticos en las dependencias policiales de Venezuela, fue desarrollado el SMW.

Debido a que es procesada considerable cantidad de información geográfica en SIGEPOL, y con el objetivo de mejorar el monitoreo, localización y control del funcionamiento de las acciones que se llevan a cabo sea más eficiente, surge la idea de integrar el SMW y SIGEPOL mediante la implementación de nuevas funcionalidades sobre el SMW que permitan procesar y representar la información geográficamente referenciada que maneja SIGEPOL.

Con la solución propuesta los funcionarios de las dependencias policiales tendrán la posibilidad de realizar con apoyo de los mapas, análisis a niveles más profundos para la toma de decisiones y estrategias a seguir ante situaciones determinadas. Pues con el fin de optimizar las prestaciones que

⁴ Centro de Tratamiento y Análisis de Información de Seguridad Ciudadana.

SIGEPOL ofrece se contribuirá a un mejor desempeño de las labores para velar por la seguridad ciudadana en Venezuela.

La información geográfica combinada con las bases de datos de los hechos delictivos permite realizar análisis espaciales y crear mapas de criminalidad, que sirven y orientan en el planteo de soluciones. Un entendimiento del “dónde” y “cuándo” los delitos ocurren, puede mejorar significativamente el rendimiento de los esfuerzos dedicados a la prevención del crimen.

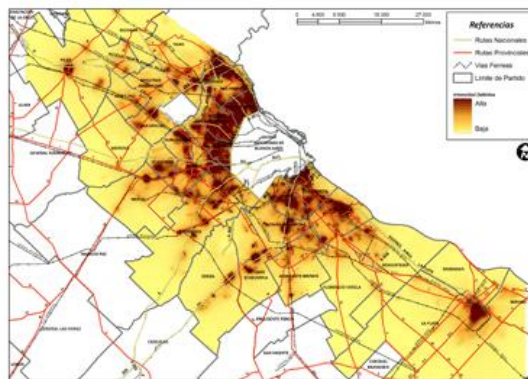


Fig. 1.1 Mapa de Intensidad Delictiva

Los mapas ofrecen a los analistas representaciones geográficas, pues el mapeo del delito ayuda a los funcionarios de las dependencias policiales a proteger, de manera más efectiva, a los ciudadanos en sus respectivas áreas de responsabilidad.

1.3 Sistema de Gestión Policial

Sistema que tiene como objetivo la captura de la información sobre la gestión policial a nivel nacional, que sirva de base a la toma de decisiones y ayude a combatir el delito en la República Bolivariana de Venezuela. (4) Este unifica y comparte la información relacionada con denuncias, reseñas y operativos policiales que se registran en las dependencias de las policías estatales y municipales para garantizar una mejor eficiencia de su trabajo.

La información relevante a la actividad de las dependencias policiales, es gestionada por el sistema informático mediante cuatro módulos principales:

- ✓ Módulo de Reseña.

- ✓ Módulo de Denuncia.
- ✓ Módulo de Operativos Policiales.
- ✓ Módulo de Dependencia.

Estos módulos se relacionan entre sí para lograr el correcto funcionamiento del sistema. A continuación, una breve descripción acerca de cada uno de ellos.

1.3.1 Módulo de Reseña

Reseñar consiste en describir características o rasgos distintivos de un determinado evento, objeto o persona. Este módulo permite a las dependencias policiales registrar reseñas mediante el almacenamiento de datos personales de los ciudadanos que se detengan por delitos o faltas cometidos. También se generan reportes sobre los ciudadanos, agrupados bajo diversos criterios, entre los que podemos citar:

- ✓ Fecha en la que se cometió el delito o falta.
- ✓ Dirección de la Detección.
- ✓ Tipo de delito o falta.

1.3.2 Módulo de Denuncia

Tramita la información correspondiente a las denuncias, las cuales consisten en la declaración de un hecho delictivo por parte de una persona que ha sido afectada por otra persona y/o entidad. Son recogidos los datos personales del autor de la denuncia en caso que no sea anónima. Además, se recoge la declaración y los datos del denunciado entre los que se encuentra su dirección, si es una persona natural, en el caso de un ciudadano, o una persona jurídica en el caso de una empresa o entidad.

1.3.3 Módulo de Operativos Policiales

Se encarga de gestionar los operativos policiales planificados por las dependencias correspondientes con el fin de combatir y prevenir el delito en un área geográfica determinada. Por cada operativo policial planeado, se definen y planifican los mecanismos de funcionamiento, y de estos se especifica el tipo de mecanismo, el período de tiempo para su ejecución, el área de acción sobre la que se ejecutará, entre

otras características. De igual forma se administra el cierre de los mecanismos correspondientes a un operativo una vez concluido el mismo, permitiendo un estudio de los resultados obtenidos para evaluar el éxito de la actividad.

1.3.4 Módulo de Dependencia

Monitorea la información referente a las dependencias policiales, tales como dirección, armamento en posesión y el área de responsabilidad de la misma. También gestiona la información de los funcionarios que pertenecen a dichas dependencias, de los mismos se controlan las armas de fuego en préstamo, expedientes abiertos, méritos, jerarquía y sanciones impuestas.

1.4 Sistema de Información Geográfica (SIG)

Históricamente, la utilización de mapas como herramientas para la toma de decisiones se remonta a varios siglos atrás. Un mapa se comprende como un modelo gráfico del territorio en el que se representan determinadas características de acuerdo a una simbología. En la actualidad existen avanzadas técnicas de cartografía digital, orientadas a la informatización de diversos tipos de mapas.

Según la NCGIA⁵ se define Sistema de Información Geográfica (SIG) como un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados para resolver problemas complejos de planificación y gestión. (5)

El objetivo principal de un SIG es la capacidad de analizar información espacial compleja, y generar nueva información a partir de un conjunto previo de datos espaciales y temáticos mediante su manipulación y reelaboración.

1.4.1 Aplicaciones de los SIG

Los SIG han constituido una verdadera revolución conceptual y práctica en el manejo de la información geográfica, siendo considerada el paso más importante desde la invención del mapa en cuanto a la utilización de los datos espaciales y por tanto muy empleada en diferentes esferas de la sociedad.

⁵ National Center for Geographic Information and Analysis.

A continuación, enunciamos algunas de las aplicaciones de los SIG:

- ✓ **SIG básico:** Gestión de infraestructura de información geográfica digital.
- ✓ **Agricultura y usos del suelo:** Inventario y simulación de cosechas.
- ✓ **Prospección:** Petróleo y minerales.
- ✓ **Cartografía:** Producción de mapas, redes de infraestructura para inventario y gestión de acueducto, alcantarillado, electricidad, gas y teléfono.
- ✓ **Análisis de fenómenos sociales:** Inventario, gestión y modelamiento.
- ✓ **Tráfico:** Control de flotas, envío de unidades de emergencia, rutas óptimas.
- ✓ **Manejo de Peligros:** Localización de áreas vulnerables, respuesta ante desastres.
- ✓ **Bélicas:** Inventario, gestión y modelamiento de información geográfica general y de interés militar. (6)
- ✓ **Instituciones de Seguridad Ciudadana:** Visualización en un área geográfica de elementos o incidentes determinados; como la ubicación de eventos que son informados al centro por organismos o personas que requieran del apoyo de los órganos de seguridad; ubicación de sitios referenciados geográficamente como estaciones de bomberos y policías, hospitales y vehículos en servicio. Puede también ser usado para obtener información crítica mientras los responsables se están dirigiendo hacia la ubicación de un incidente determinado, dando asistencia en el planeamiento táctico y respuesta. (7)

En nuestro país existe el grupo empresarial GEOCUBA, creador del SIG para el manejo integral del Ecosistema Sabana Camagüey. Las universidades no se han quedado atrás en lo referente al desarrollo de estos sistemas; un ejemplo lo constituye la CUJAE⁶, donde se desarrolló un SIG para la gestión de las estadísticas de salud para nuestro país. Otra muestra de la utilización de SIG se encuentra presente en la Universidad de las Ciencias Informáticas, denominado Sistema de Información Geográfica UCI (SIGUCI), ofreciendo la posibilidad de representar visualmente todo tipo de información espacial relevante acerca de la universidad.

⁶ Instituto Superior Politécnico "José Antonio Echeverría".

1.4.2 Sistema de Mapificación Web

El SMW ofrece la capacidad de procesar información referenciada geográficamente, para ser utilizado por las dependencias policiales de la República Bolivariana de Venezuela. (1) Para el desarrollo de la solución propuesta las nuevas funcionalidades para la representación y procesamiento de información geográficamente referenciada de SIGEPOL se implementan sobre el SMW.

1.4.2.1 Características Generales

El SMW tiene la capacidad de procesar datos tanto vectoriales como ráster. El procesamiento de datos vectoriales se realiza mediante la descripción de los objetos geográficos por medio de vectores, muy útil para la representación de características discretas, donde la información es representada a través de puntos, líneas y polígonos almacenados como una colección de coordenadas (x, y). A estos objetos de dibujo se les puede asociar las diversas capas de información que se relacionan con el modelo espacial generado a través de puntos y líneas. (1)

Para el manejo de objetos geográficos con límites difusos o de características muy variables se utiliza la categoría ráster. Este modelo utiliza una malla rectangular de pequeñas celdas, denominadas píxel. Cada píxel recibe un número como representación a su valor temático. Este número porta la información necesaria para modelar un aspecto del medio. Dado que la malla es regular (el tamaño del píxel es constante) y que conocemos la posición en coordenadas del centro de una de las celdas, se puede decir que todos los píxeles están georreferenciados. (1) Ejemplos de este tipo de datos espacial lo constituyen las imágenes satelitales, generalmente en formato jpeg o tiff.

Entre las funcionalidades que posee el sistema se encuentran las operaciones básicas de navegación sobre mapa tales como insertar un elemento, acercar mapa, alejar mapa, centrar mapa, manipular capa. Permite también seleccionar regiones determinadas, insertar una línea así como eliminar un elemento especificado.

La tecnología SIG utilizada en el desarrollo del SMW fue MapInfo MapXtreme Java Edition 4.8, debido a sus características tales como ser multiplataforma, compatible con varios servidores web y de aplicación como Apache Tomcat y el servidor de aplicación Oracle respectivamente, de alto nivel de configuración y rendimiento.

1.4.2.2 Arquitectura

La propuesta de arquitectura del SMW se corresponde con el patrón arquitectónico Layers (Capas). Está conformada por objetos del Dominio y las capas de Presentación, Negocio y Acceso a Datos. La utilización de frameworks especializados permite realizar abstracciones y definir de una forma precisa la separación entre las capas. (1)

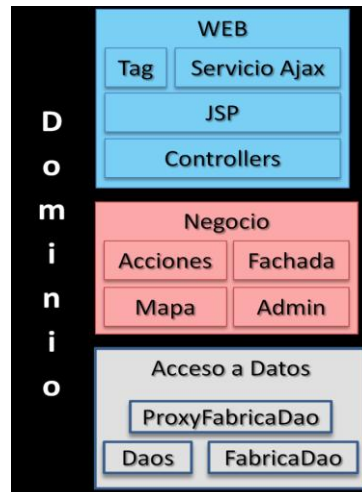


Fig. 1.2 Arquitectura SMW

1.4.2.3 Valoración del Diseño de la Arquitectura

El presente diseño está basado sobre una arquitectura en capas que potencia la estructuración de cada capa lógica agrupando las clases de acuerdo a sus responsabilidades. Sin embargo, no es la mejor opción a utilizar para la solución propuesta dada su poca extensibilidad, pues imposibilita la separación de las funcionalidades básicas ya existentes del SMW de las nuevas funcionalidades especificadas para el procesamiento y visualización de la información geográficamente referenciada de SIGEPOL.

1.5 Tecnología SIG a utilizar

La tecnología SIG permite almacenar y manipular información geográfica, mediante el análisis de patrones, relaciones, y tendencias en la información, con el objetivo de mejorar la toma de decisiones.

Actualmente cerca del 80% de la información tratada por instituciones y empresas públicas o privadas tienen en alguna medida relación con datos espaciales, siendo de gran importancia la toma de medidas pues la misma depende en gran parte de la calidad, exactitud y actualidad de la información espacial.

1.5.1 MapInfo MapXtreme Java Edition

MapInfo MapXtreme Java es un servidor de gestión de mapas 100% Java diseñado para permitir el uso de inteligencia de localización en Internet, en la intranet o extranet de una organización. Diseñado para organizaciones que buscan un despliegue amplio de mapas y análisis basado en localización, MapXtreme Java ofrece a los desarrolladores de aplicaciones un componente intuitivo y altamente visual para integrar funcionalidad de mapas en cualquier aplicación web. (8)

Cumple con la especificación J2EE y provee a los desarrolladores una gran flexibilidad. Permite ejecutar aplicaciones de mapas en un servidor centralizado reduciendo los costos de hardware y administración mientras mejora el desempeño, confiabilidad y escalabilidad de la aplicación.

1.6 Plataforma de Software a utilizar

Una plataforma de software constituye el ambiente de hardware o en el caso de software un conjunto de librerías y programas en el cual se ejecutan aplicaciones. Un ejemplo de plataforma lo constituye un entorno de programación, sistema operativo, o generalmente la composición de ambos.

La plataforma Java es independiente de la plataforma de hardware al igual que .NET, plataforma de Microsoft para su sistema operativo Windows; pero la primera tiene una mayor ventaja, pues mediante el uso de una máquina virtual autónoma es capaz de leer el código intermedio o como también se le conoce bytecode, sin necesidad de la interferencia del sistema operativo en que se ejecuta.

1.6.1 La Plataforma Java

Entorno de desarrollo procedente de Sun Microsystems. Constituye una plataforma virtual que provee:

- ✓ Lenguaje de programación Java.

- ✓ Interfaz de Programación de Aplicaciones (API) ⁷.
- ✓ Máquina Virtual de Java con su propio conjunto de instrucciones.

Es la plataforma ideal para la informática de la red. Corre sobre diversos tipos de plataformas, desde servidores hasta teléfonos móviles y tarjetas inteligentes. La misma proporciona un compilador para traducir el código fuente al conjunto de instrucciones de la máquina virtual de Java. Posee un intérprete para ejecutar las instrucciones de la máquina virtual y una implementación de la API de Java. (9)

La plataforma Java ha tenido gran aceptación a nivel mundial y hasta el momento ha fascinado a más de seis millones y medio de desarrolladores de software. Es utilizada en ámbitos tan diferentes como el académico o el de grandes proyectos corporativos, así como en los principales sectores de la industria y se encuentra en un gran número de dispositivos, equipos y redes. Pues es una plataforma basada únicamente en software que corre por encima de las plataformas basadas en hardware. (10)

1.7 Frameworks a utilizar

Framework, marco de trabajo o plataforma en cuanto a desarrollo de software que se define como estructura conceptual y tecnológica de soporte definido; donde un proyecto de software puede ser organizado y desarrollado. Puede contener soporte para programas, bibliotecas y/o un lenguaje interpretado. Permite a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel que provee un sistema funcional.

1.7.1 Ext JS

Ext JS es una librería de JavaScript ligera, de alto rendimiento para el desarrollo rápido de aplicaciones web cross-browser. Presenta una interfaz de usuario personalizable “widgets”, bien diseñado y contiene un modelo de componentes extensibles. Su API es intuitiva y fácil de usar. (11)

Tiene disponible licencias comerciales y de código abierto. Es compatible con la mayoría de los navegadores, entre los cuales se puede citar:

- ✓ Internet Explorer 6 +.

⁷ Application Programming Interface.

- ✓ FireFox 1.5 +.
- ✓ Opera 9 +.

La familia de productos de Ext es usada por miles de compañías a lo largo del mundo en diversas industrias, entre las que se encuentran:

- ✓ Adobe.
- ✓ IBM.
- ✓ CISCO.
- ✓ SONY.
- ✓ Pixar Animation Studios.

1.7.2 Spring Framework

Spring, es un framework de código abierto para el desarrollo de aplicaciones, existen versiones disponibles para la plataforma Java y .NET. Spring ofrece un diseño y arquitectura sólido y confiable, basados en las mejores prácticas y patrones de programación aceptados en la actualidad. A pesar de que las principales funcionalidades de Spring puede ser utilizadas en cualquier programa Java, cuenta con diversas mejoras y extensiones para mejorar en gran medida el desarrollo de aplicaciones web en plataforma empresarial de Java.

Spring 3.0 está constituido por funcionalidades organizadas en 20 módulos que a su vez se agrupan en 7 contenedores, capaces de proveer una extensa gama de funcionalidades de fácil uso. (12) La flexibilidad de su diseño e implementación permite seleccionar y trabajar con los módulos necesarios para el desarrollo de la aplicación e ignorar el resto. Seguidamente se muestran los contenedores utilizados:

Contenedor Núcleo: Contiene la clase BeanFactory que es el corazón de toda aplicación sobre Spring. Aplica la Inversión de Control (IoC)⁸ y la Inyección de Dependencia, para separar la configuración de la aplicación y sus especificaciones de dependencias entre objetos de la lógica de negocio de la aplicación. Contiene el módulo Lenguaje de Expresión, potente en cuanto a la consulta y manipulación de objetos en tiempo de ejecución.

⁸ Inversion of Control

Contenedor Integración/Acceso a Datos: Aquí se brinda soporte a las funcionalidades y problemáticas comunes en el trabajo con la base de datos. Permiten abstraer al programador de las acciones básicas de intercambio con la base de datos siendo posible mantener el código de acceso a datos sencillo y legible. Implementa mecanismos de comunicación para lograr compatibilidad e integración con otros frameworks que soportan el mapeo de objetos relacionales como iBatis e Hibernate.

Contenedor Web: Útil para el desarrollo de aplicaciones web basando el intercambio del cliente con el servidor, en la filosofía que sigue el patrón arquitectónico nombrado Modelo-Vista-Controlador (MVC)⁹, donde provee una separación entre el dominio, el modelo y los controladores. Inicializa el contenedor IoC utilizando servlets escucha y permite unir declarativamente parámetros de una petición a objetos de negocio en un controlador especificado.

Seguidamente se muestran algunas de las nuevas funcionalidades que ofrece la versión 3.0 del framework Spring a utilizar en el desarrollo de la solución propuesta:

Modelo de estereotipos: Permite crear anotaciones de “atajo” a través del uso de meta-anotaciones. Por ejemplo, para determinar ámbitos predeterminados o características de transacción predeterminadas en estereotipos propios.

Anotaciones: Provee un soporte para la especificación de Inyección de Dependencia en Java mediante anotaciones. Se define la anotación auto enlace o Autowire en inglés para la inyección de dependencia de componentes directamente en el constructor de la clase que se especifique. Se encuentra disponible la detección automática de componentes de Spring. Estas características contribuyen en gran medida a la reducción del número de configuraciones basadas en XML innecesariamente.

Soporte para controladores basados en anotaciones: Permite anotar a clases como controladores, desaparece la necesidad de heredar de clases abstractas e implementar interfaces. También propone un nuevo modelo de controlador web para el mapeo de peticiones a métodos anotados, ventaja que permite implementar controladores Multi-Action con mayor facilidad.

⁹ Model-View-Controller.

1.7.3 Ibatis

Framework de código abierto que facilita la comunicación de una base de datos relacional con aplicaciones Java y .NET. Ibatis mapea directamente objetos con procedimientos almacenados, o sentencias SQL mediante ficheros XML. La simplicidad es la mayor ventaja de Ibatis sobre las demás herramientas de este tipo y cuenta con una amplia documentación.

Se encuentra estructurado por dos paquetes complementarios pero independientes: Ibatis Data Access Objects (DAO), que implementa la capa de abstracción e Ibatis SQL MAPS, que implementa la capa de persistencia. (13) Ibatis Data Access Objects (DAO) oculta los detalles de la capa de persistencia y proporciona un API común para todas las aplicaciones.

1.8 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML)¹⁰ constituye un lenguaje que permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Se compone de diversos elementos gráficos que se combinan para conformar diagramas. Este lenguaje dispone de reglas para combinar tales elementos y permite la modelación de sistemas con tecnología orientada a objetos. Los diagramas son entes importantes de UML, cuya finalidad es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, no cómo implementarlo. El modelo gráfico de UML tiene un vocabulario en el que se identifican: elementos, relaciones y diagramas.

1.9 Metodología de Desarrollo de Software

Metodología de desarrollo de software según Jacobson y Rumbaugh, es el conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

Cuando se decide realizar un producto de software es muy importante especificar qué metodología de desarrollo de software emplear, como guía de apoyo para lograr un correcto desenvolvimiento, control del

¹⁰ Unified Modeling Language.

proyecto y un producto que satisfaga las necesidades del cliente. Para la selección de esta metodología se debe tener en cuenta la duración del proyecto, así como la frecuencia de los cambios a los que este puede estar sujeto, pues no existe una metodología de desarrollo de software universal. Por lo que las características de cada proyecto exigen que el proceso sea configurable.

De las distintas metodologías de desarrollo de software que existen se encuentran:

- ✓ Programación Extrema (XP)¹¹.
- ✓ Desarrollo Guiado por Funcionalidad (FDD)¹².
- ✓ Proceso Unificado de Desarrollo (RUP)¹³.

1.9.1 Metodología Proceso Unificado de Desarrollo (RUP)

RUP constituye el resultado de años de aplicación en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes.

Su modelación es definida mediante Quién (trabajadores) debe hacer Qué (artefactos), Cuándo (Flujo de actividades) y Cómo (actividades) debe hacerlo (14).

En RUP existen 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como flujos de apoyo. En la siguiente figura se representa el proceso en el que se grafican los flujos de trabajo, las fases del desarrollo de software y la dinámica expresada en iteraciones y puntos de control.

¹¹ Xtreme Programming.

¹² Feature Driven Development.

¹³ Rational Unified Process.

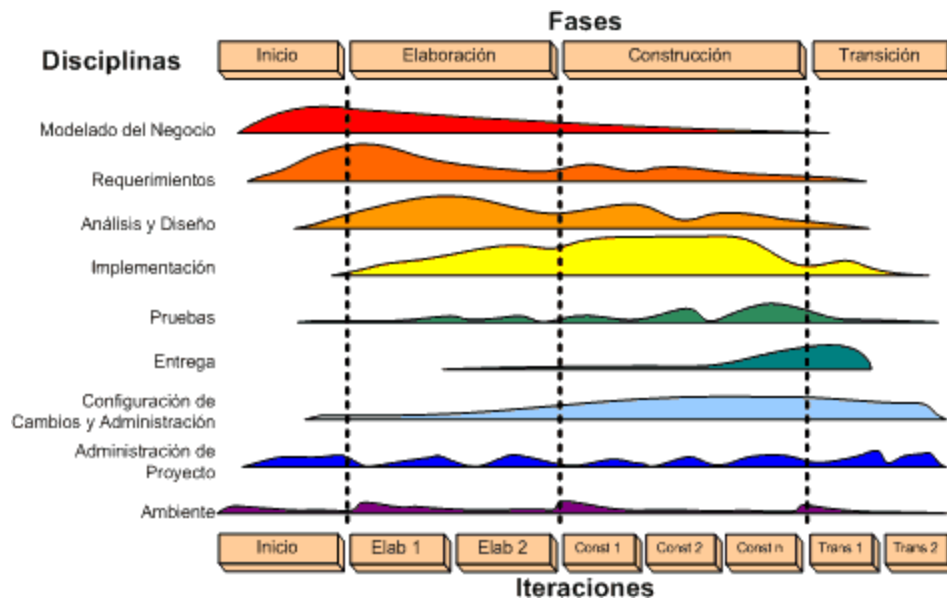


Fig. 1.3 RUP en dos dimensiones

Fases:

- ✓ **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo el alcance con la identificación de los casos de uso del sistema.
- ✓ **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- ✓ **Construcción:** Se obtiene un producto listo para su utilización que está documentado y posee un manual de usuario. Se obtiene uno o varias entregas del producto que han pasado las pruebas.
- ✓ **Transición:** La entrega del producto está lista para su instalación en las condiciones reales.

1.10 Herramientas de Desarrollo

Las herramientas de desarrollo de software son de vital importancia para la realización y control del desarrollo de una solución de software. De acuerdo a su objetivo presenta diversas opciones de gran utilidad permitiendo que el proceso de producción del software sea ágil y eficaz. Por ejemplo en el caso de los entornos de desarrollo, la generalidad cuenta con auto completamiento de código, detección de errores; así como también incluyen códigos de ejemplo, notas técnicas y documentación de soporte.

1.10.1 Eclipse

Eclipse es un ambiente de desarrollo integrado (IDE)¹⁴ multiplataforma y de código abierto basado en Java. Fue desarrollado originalmente por IBM. Fue creada la comunidad Eclipse, organización autónoma que promueve el código abierto y una serie de servicios y productos suplementarios. (15)

Lo exclusivo de Eclipse es la extensibilidad, pues es una estructura integrada por un núcleo y muchos plug-ins conformando la funcionalidad final. Los plug-ins interactúan mediante interfaces o puntos de extensión; evitando problemas de conflicto o integración al entorno.

En el perfeccionamiento del SMW fueron utilizados un conjunto de plug-ins que amplían las funcionalidades de la plataforma de desarrollo de aplicaciones sobre la web. Los plug-ins más importantes son los siguientes:

- ✓ **Web Tool Platform (WTP):** Para soportar todas las funciones necesarias para desarrollar aplicaciones web sobre J2EE.
- ✓ **Spring IDE:** Para facilitar el uso sobre los beans y xml definidos por Spring Framework.
- ✓ **Subclipse:** Para permitir de una forma mucho más ágil y cómoda el desarrollo colaborativo de software en el equipo de desarrolladores mediante la comunicación con un repositorio subversion.

¹⁴ Integrated Development Environment.

1.10.2 Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Ordenador (CASE)¹⁵, son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo del producto de software, reduciendo el coste en términos de tiempo y dinero. (16)

Ayudan a organizar y manejar la información de un proyecto informático. Permite que los sistemas principalmente de alta complejidad, como es el caso de SIGEPOL, se tornen más flexibles y comprensibles para lograr un mejor entendimiento por parte de sus desarrolladores.

1.10.2.1 Visual Paradigm

Visual Paradigm es una eficaz herramienta proveniente del líder de soluciones de software Visual Paradigm International, que facilita a las organizaciones diseñar mediante diagramas e integrar y desplegar las funcionalidades críticas de aplicaciones empresariales.

Es un producto de alta calidad donde el modelado de sistemas y soluciones de software se realiza utilizando UML, permite crear diagramas de clases y artefactos que tributan a la documentación de un software a lo largo del proceso de desarrollo. Se integra fácilmente con varios IDEs de programación como Eclipse y Visual Studio .Net. Permite la generación semiautomática de código a partir de los diagramas construidos y también se puede llevar a cabo la ingeniería inversa para refinar los modelos. Soporta aplicaciones web y existe compatibilidad entre sus ediciones. Acelera el desarrollo del software obteniendo productos de calidad reduciendo costos y riesgos. (17)

1.12 Conclusiones

Durante el presente capítulo, fue expuesta la fundamentación teórica donde se argumenta el problema científico a resolver, así como fue explícitamente descrita la estructura funcional de SIGEPOL y el SMW, información necesaria para el entendimiento de la solución a proponer y base para el desarrollo de la misma, dándose cumplimiento al primer objetivo específico del trabajo de diploma.

¹⁵ Computer Aided Software Engineering

Se reafirma la elección de las tecnologías y herramientas de desarrollo a emplear en la mejora de la aplicación, pues en la presente tesis se implementan nuevas funcionalidades sobre el SMW, tesis de pregrado del Ing. Reynier González Tejeda, de manera tal que el SMW quede adaptado a las necesidades de SIGEPOL gestionando la información geográficamente referenciada de este último.

La tecnología SIG a utilizar para el perfeccionamiento del SMW es MapInfo MapXtreme Java Edition 4.8. Como metodología de desarrollo de software se trabajará con RUP y la herramienta CASE a utilizar será Visual Paradigm UML for Enterprise Edition versión 6.4. La aplicación residirá sobre la plataforma Java. Los frameworks a utilizar serán Spring 3.0, Ibatis 2.3, y Ext JS 2.2.1, y como IDE se empleará el Eclipse Galileo.

Capítulo 2. Características del Sistema

2.1 Introducción

En el presente capítulo se describe la nueva propuesta del SMW. Se muestra el modelo de dominio mediante la representación de los principales conceptos identificados. Se enumeran los requerimientos funcionales así como los requerimientos no funcionales; constituyendo las capacidades a cumplir por el sistema y las cualidades que este producto debe tener respectivamente. Se identifican los actores, casos de uso del sistema así como su respectiva descripción, utilizando como metodología de desarrollo RUP y UML como lenguaje de modelado.

2.2 Propuesta del Sistema

A continuación, se describe de manera general la nueva propuesta del SMW y como éste debe funcionar.

2.2.1 Descripción General

Las nuevas funcionalidades a desarrollar en el SMW darán cumplimiento a la necesidad de SIGEPOL de procesar y representar la información geográfica que maneja de manera visual en un mapa.

El usuario analista información geográfica ingresa en el sistema, y la primera vista que se muestra es un mapa con la ubicación de las dependencias policiales. El usuario podrá realizar búsquedas para conocer la ubicación geográfica de hechos delictivos y detenciones que han sucedido en un área determinada según los criterios especificados. Podrá también buscar la localización de las dependencias policiales y visualizar sus respectivas zonas de responsabilidad. El sistema brindará la posibilidad de mostrar detalles específicos por cada elemento mostrado en el mapa.

Mediante la solución propuesta los usuarios que utilicen el sistema podrán realizar análisis profundos para la toma de decisiones ante determinadas situaciones. El SMW quedará adaptado a los requerimientos de SIGEPOL ofreciendo servicios de procesamiento de la información geográfica presente en este último. Con el desarrollo de la solución se contribuye a un mayor control de las acciones que se llevan a cabo para contrarrestar incidentes delictivos, pues mediante el conocimiento de “dónde” y

“cuándo” los delitos ocurren pueden ser identificados patrones de comportamiento, optimizando significativamente el rendimiento de los esfuerzos dedicados a la prevención del crimen.

2.3 Modelo del Dominio

El modelo del dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Facilita en gran medida la comprensión del problema mediante un vocabulario común a los usuarios, clientes, desarrolladores e interesados en el desarrollo del sistema. Muestra la relación entre los principales conceptos y contribuye a una correcta y eficiente captura de requisitos.

Puesto que no existe experiencia de trabajo con SIG sobre Web en SIGEPOL y que los procesos en este sentido no se encuentran bien identificados, se considera entonces un escenario apropiado para aplicar un modelo de dominio proponiéndose su realización.

2.3.1 Conceptos del Modelo de Dominio

- ✓ **Usuario:** Actor que interactúa con el SIG (analista información geográfica).
- ✓ **SMW:** Sistema capaz de integrar, almacenar, gestionar y representar en pantalla información geográfica referenciada.
- ✓ **Capa:** Colección de elementos que poseen el mismo conjunto de atributos.
- ✓ **Elemento:** Es la unidad de información básica de una capa. Se clasifica en punto, línea y polígono.
- ✓ **Punto:** Consiste en el par coordenado (latitud, longitud) o (x, y) del SIG donde se mostrará la imagen que puede representar una dependencia policial, un hecho delictivo o una detención.
- ✓ **Línea:** Constituye una serie ordenada de posiciones o puntos unidos por segmentos rectos. Se utilizan para modelar calles, avenidas y ríos.
- ✓ **Polígono:** Compuesto por líneas cerradas que delimitan superficies. Modelan parroquias, municipios, estados o el área de cobertura correspondiente a una dependencia policial determinada.

✓ **Mapa:** Conjunto de capas superpuestas.

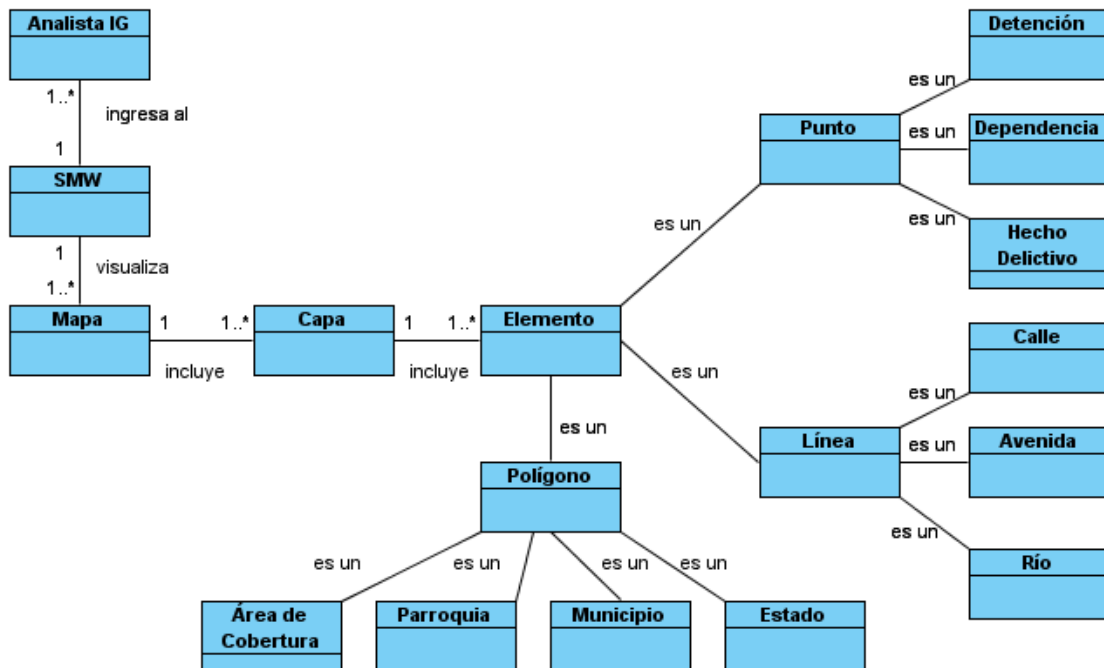


Fig. 2.1 Modelo de Dominio

2.3.2 Descripción del Modelo de Dominio

Los usuarios que interactúan con el SMW podrán acceder a todas las funcionalidades definidas. La aplicación utiliza un mapa que está constituido por varias capas y cada una por un conjunto de elementos de una misma geometría, es decir, cada capa está compuesta por elementos de una misma categoría.

El SMW permite realizar operaciones básicas de navegación sobre mapas; pero con el desarrollo de las nuevas funcionalidades se podrá mostrar información geográfica relacionada a cada dependencia policial, como las detenciones realizadas en su área de cobertura correspondiente y hechos delictivos ocurridos, así como es posible la consulta de información detallada de estos elementos

2.4 Requerimientos

Se enuncian las capacidades o condiciones que el sistema debe cumplir como requerimientos funcionales y constituyen el punto de partida para identificar qué debe hacer el sistema. (14)

Los requerimientos no funcionales son aquellos que especifican qué cualidades o propiedades que el producto debe tener. Ambos requerimientos deben ser comprendidos por los desarrolladores y los usuarios.

2.4.1 Definición de Requerimientos Funcionales

A continuación, los requerimientos funcionales que se deben cumplir en el perfeccionamiento del SMW:

RF1. Autenticar Usuario

Autentica al usuario, solicitando el usuario y la contraseña:

- ✓ Verificar que el usuario esté registrado en la base de datos de SIGEPOL.
- ✓ Verificar que la contraseña sea correcta.

RF2. Manipular Capas

Permite mostrar/ocultar el contenido de las capas del mapa.

RF3. Mostrar Leyenda

Visualiza la simbología de los elementos mostrados en el mapa.

- ✓ Icono del elemento.
- ✓ Descripción.

RF4. Mostrar Dependencias Policiales

Muestra la ubicación geográfica de cada dependencia policial en el mapa al iniciar el sistema.

RF5. Visualizar Dependencias Policiales

Permite mostrar la ubicación geográfica de las dependencias policiales seleccionadas, que cumplan los criterios de búsqueda siguientes:

- ✓ Nombre Dependencia.
- ✓ Código Dependencia.
- ✓ Organismo Superior.
- ✓ Tipo Dependencia.
- ✓ Estado.
- ✓ Municipio.
- ✓ Parroquia.
- ✓ Localidad.

RF6. Mostrar Detalle de una Dependencia Policial

Muestra información relevante de la dependencia policial seleccionada.

- ✓ Nombre Dependencia.
- ✓ Código Dependencia.
- ✓ Tipo Dependencia.
- ✓ Organismo Superior.
- ✓ Descripción.
- ✓ Teléfonos.
- ✓ Email.
- ✓ Fax.

RF7. Visualizar Hechos Delictivos

Permite visualizar en el mapa los hechos delictivos seleccionados, que cumplan los siguientes criterios de búsqueda:

- ✓ Código Dependencia.
- ✓ Fecha Inicial.
- ✓ Fecha Final.

- ✓ Delito/Falta.
- ✓ Estado.
- ✓ Municipio.
- ✓ Localidad.
- ✓ Parroquia.

RF8. Mostrar Detalle de un Hecho Delictivo

Permite mostrar la siguiente información sobre un hecho delictivo determinado:

- ✓ Código Dependencia.
- ✓ Nombre Dependencia.
- ✓ Modus Operandi.
- ✓ Fecha Inicial.
- ✓ Fecha Final.
- ✓ Delito y/o falta.
- ✓ Cantidad de denuncias asociadas a ese hecho delictivo.

RF9. Visualizar Detenciones

Permite visualizar la ubicación geográfica de aquellas detenciones, que cumplan los siguientes criterios de búsqueda:

- ✓ Código Dependencia.
- ✓ Nombre Dependencia.
- ✓ Número de Detención.
- ✓ Fecha Inicial.
- ✓ Fecha Final.
- ✓ Estado.
- ✓ Municipio.
- ✓ Localidad.
- ✓ Parroquia.

RF10. Mostrar Detalle de una Detención

Muestra la información relevante correspondiente a la detección seleccionada. Los criterios son:

- ✓ Código Dependencia.
- ✓ Nombre Dependencia.
- ✓ Número de Detención.
- ✓ Fecha de la Detención.
- ✓ Rango de Hora.
- ✓ Cantidad de Detenidos.

RF11. Mostrar Áreas de Cobertura de Dependencias Policiales

Visualiza en el mapa el área de cobertura o responsabilidad de las dependencias especificadas.

2.4.2 Definición de Requerimientos No Funcionales

Los requerimientos no funcionales son las propiedades o cualidades que permiten al producto ser usable, atractivo, rápido y confiable. Generalmente influyen en el éxito y aceptación del producto por el cliente. A continuación, se proponen los requerimientos no funcionales que debe tener el sistema.

Requerimientos de Interfaz

Interfaz de Usuario:

La interfaz de usuario del sistema debe ser amigable y debe cumplir los requisitos siguientes:

- ✓ La estructura general del sistema debe estar bien definida y la información que se muestre debe ser fácil de entender.
- ✓ La interfaz debe contar con opciones que faciliten la utilización del sistema y con una leyenda que visualice la simbología del mapa, según los elementos que se muestren.
- ✓ Utilizar una norma de diseño para la representación de los elementos de las ventanas, es decir, colores, formato de la fuente y formato de elementos a representar visualmente.

- ✓ Validar la entrada de información del usuario, e identificar los errores que se detecten para que el usuario los corrija.
- ✓ Utilizar iconos y/o estilos para identificar cada elemento en el mapa.
- ✓ Toda la información a mostrar por el sistema debe aparecer en idioma español.

Interfaz de Comunicación:

- ✓ La comunicación entre el servidor de la base de datos de SIGEPOL y el SMW es mediante el protocolo TCP/IP.
- ✓ El servidor de gestión de mapas MapInfo MapXtreme y el SMW se comunican utilizando el protocolo http por el puerto 8080.

Requerimientos de Soporte

Garantizar la instalación y prueba del sistema, así como proveer la preparación e instrucción necesaria a los usuarios que trabajarán con la aplicación.

Requerimientos de Portabilidad

El SMW debe ser multiplataforma para garantizar que puede ser utilizado en distintos sistemas operativos.

Requerimientos de Seguridad

Confidencialidad: La información del sistema solo podrá ser consultada por el personal autorizado.

Disponibilidad: El sistema debe estar disponible las 24 horas del día.

Integridad: El sistema debe ser de carácter informativo, es decir, el usuario solo puede realizar búsquedas de la información deseada y no modificar la misma.

Requerimientos Legales

El SMW y la documentación generada correspondiente pertenecen al proyecto SIGEPOL.

Requerimientos de Confiabilidad

- ✓ El tiempo medio para la reparación de fallas debe ser como máximo un día.

Requerimientos de Software

Para todas las terminales:

- ✓ Debe de tener instalado un navegador web (Mozilla Firefox versión 3.6 o Internet Explorer versión 7.0).

Terminal Servidor:

- ✓ Debe estar instalado el servidor web Apache Tomcat.
- ✓ Debe encontrarse instalada la máquina virtual de java correspondiente al sistema operativo en uso, a partir de la versión 1.4 en adelante.
- ✓ Debe estar instalado el servidor de gestión de mapas MapInfo MapXtreme 4.8.

Requerimientos de Hardware

Para todas las terminales:

- ✓ Todos los monitores deben ser al menos de 14 pulgadas.
- ✓ Tarjeta de video que soporte como mínimo la resolución de 1024x768 pixeles.

Terminal Servidor:

- ✓ Procesador de 3 GHz.
- ✓ Memoria RAM de 1Gb o más.
- ✓ Capacidad de almacenamiento del disco duro mínima de 20Gb.

Terminales clientes:

- ✓ CPU Pentium IV.
- ✓ Capacidad de almacenamiento del disco duro con al menos 50 Mb libres.

- ✓ Memoria RAM 256 Mb como mínimo.

Requerimientos de Diseño e Implementación

Herramientas de desarrollo:

- ✓ MapInfo MapXtreme Java Edition 4.8.
- ✓ IDE Eclipse Galileo Java EE 2009.
- ✓ Visual Paradigm for UML Enterprise Edition 6.4.

Lenguajes de programación a utilizar:

- ✓ Java.
- ✓ Javascript.
- ✓ XML (Extensible Markup Language).
- ✓ HTML (Hypertext Markup Language).
- ✓ CSS (Cascading Style Sheets).

Especificaciones del Mapa a utilizar:

Para garantizar el correcto funcionamiento del sistema, el mapa debe tener los archivos que se muestran a continuación:

Extensiones de ficheros por capa:

- ✓ MapInfo (*.TAB, *.MAP, *.ID, *.DAT).

Extensión del fichero de configuración del mapa:

- ✓ Map Definition Format (*.MDF).

El mapa a utilizar debe tener las capas siguientes:

- ✓ Estado.
- ✓ Municipio.

- ✓ Parroquia.
- ✓ Localidad.
- ✓ Calle.

2.5 Modelo del Sistema

El Modelo de Casos de Uso del Sistema especifica los actores con quien interactúa la aplicación, los casos de uso del sistema y las relaciones que existen entre ellos. Representa un esquema donde se recopilan las funcionalidades a automatizar y se determina cómo será utilizado desde el punto de vista del usuario (Actor), pues se construye sobre la base de sus necesidades.

2.5.1 Actores del Sistema

Los actores representan a los usuarios que intercambian información con el sistema. Estos suelen corresponderse con trabajadores o actores del negocio. El actor definido para la interoperabilidad con en el SMW se especifica a continuación:

Actor	Justificación
Analista Información Geográfica	Usuario autenticado en el sistema, consulta informaciones geográficas de las dependencias policiales, tales como ubicación de hechos delictivos y detenciones. Puede realizar todas las operaciones definidas en el sistema.

Tabla 2.1 Actores del Sistema

2.5.2 Paquetes

Para proveer una mejor comprensión y organización del sistema propuesto los CUS fueron agrupados en paquetes de acuerdo a las funciones que realizan:

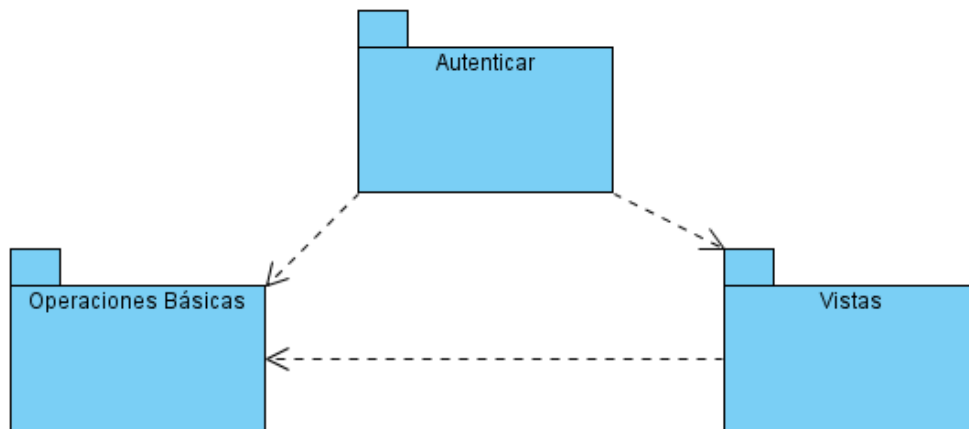


Fig. 2.2 Diagrama de Paquetes

2.5.3 Diagramas de Casos de Uso del Sistema

A continuación, se muestran los diagramas de CUS correspondiente a los paquetes presentados anteriormente.

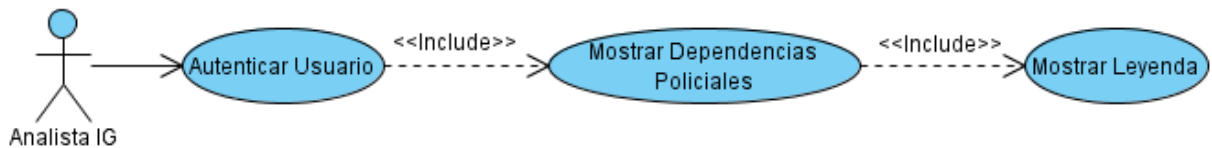


Fig. 2.3 Diagrama de CUS Paquete Autenticar

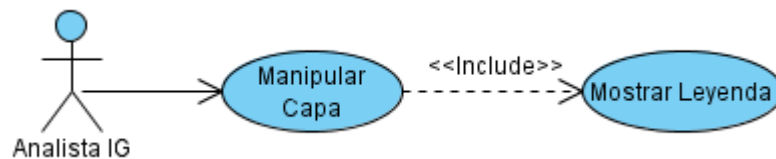


Fig. 2.4 Diagrama de CUS Paquete Operaciones Básicas

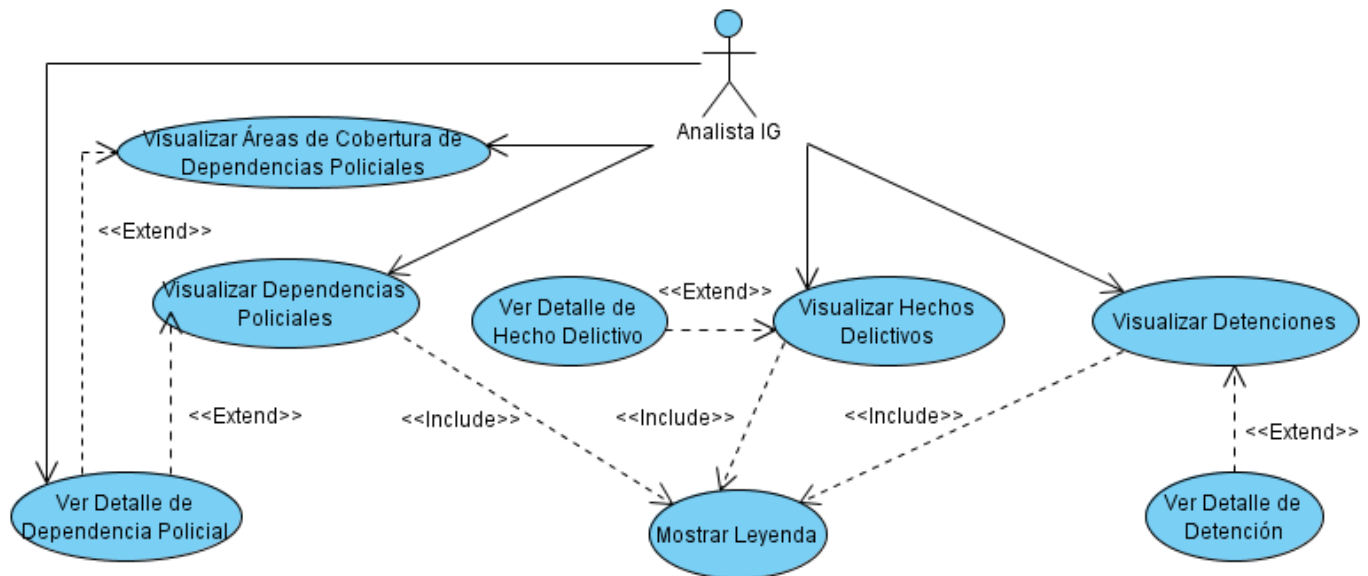


Fig. 2.5 Diagrama de CUS Paquete Vistas


2.5.4 Descripción de los Casos de Uso

Para alcanzar un mejor entendimiento de la funcionalidad asociada a cada caso de uso se especifican las respectivas descripciones a continuación.

2.5.4.1 DCUS Autenticar Usuario

Caso de Uso	Autenticar Usuario
Actores	Analista Información Geográfica
Resumen	El Analista Información Geográfica introduce los datos para autenticarse, se verifica que los datos introducidos son correctos y que el Analista Información Geográfica cuenta con los permisos necesarios para autenticarse en el sistema. En caso contrario no se permite la autenticación del usuario.
Precondiciones	Debe existir conexión con la BD.
Referencias	RF1

Prioridad	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Analista Información Geográfica introduce el usuario y la contraseña.	
2. El Analista Información Geográfica selecciona la opción "Aceptar".	3. Comprueba que existe conexión con la BD.
	4. Comprueba que los campos no estén vacíos.
	5. Comprueba en la BD que el Analista SIG se corresponde con un usuario del sistema y verifica que la contraseña pertenezca a ese usuario.
	6. Invoca al CU Mostrar Dependencias Policiales .
	7. Invoca al CU Mostrar Leyenda .
Flujo Alternativo: 3a No Existe Conexión con la BD	
Acción del Actor	Respuesta del Sistema
	3a.1) Comprueba que no existe conexión con la BD.
	3a.2) Muestra un mensaje indicando que no existe conexión con la BD.
3a.3) El usuario selecciona la opción "Aceptar".	3a.4) Cierra la aplicación.
Flujo Alternativo: 4a) Campos Vacíos	
Acción del Actor	Respuesta del Sistema
	4a.1) Comprueba que los campos usuario y/o contraseña están vacíos.
	4a.2) Indica los campos que están vacíos.

	Ir a la acción 1.
Flujo Alternativo: 5a Datos Incorrectos.	
Acción del Actor	Respuesta del Sistema
	5a.1) Comprueba en la BD que los datos de autenticación no son correctos.
	5a.2) El sistema muestra un mensaje de error indicando que la contraseña o el usuario no son válidos. Ir a la acción 1.
Poscondiciones	Usuario autenticado.
Prototipo de interfaz de usuario	
	

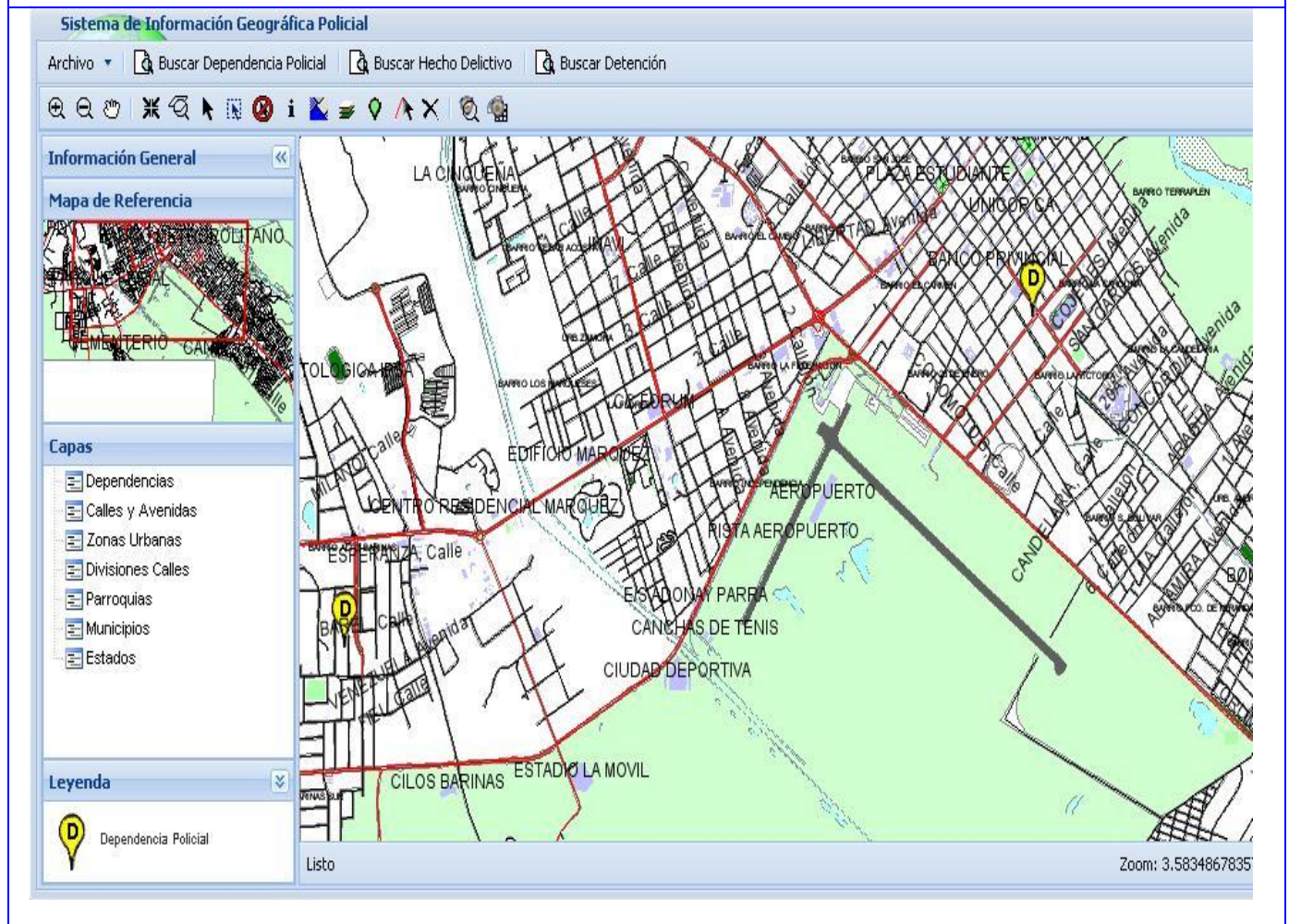
2.5.4.2 DCUS Mostrar Dependencias Policiales

Caso de Uso	Mostrar Dependencias Policiales
Actores	
Resumen	El sistema obtiene de la BD todas las dependencias policiales. Selecciona a que capa corresponden y muestra la información obtenida.
Precondiciones	<ul style="list-style-type: none"> ✓ Usuario autenticado. ✓ Debe existir conexión con la BD.
Referencias	RF1, RF4

Prioridad	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. Obtiene de la BD todas las Dependencias Policiales.
	2. Comprueba si las direcciones obtenidas de las Dependencias Policiales están en el mapa.
	3. Busca si la capa correspondiente a las dependencias obtenidas ha sido creada.
	4. Inserta el ícono que representa las Dependencias Policiales en su dirección correspondiente.
	5. Muestra en el mapa el ícono de la dependencia policial obtenida.
	6. Invoca al CU Mostrar Leyenda .
Flujo Alternativo 2a) Direcciones Incorrecta	
Acción del Actor	Respuesta del Sistema
	2a.1) Muestra el mapa.
Flujo Alternativo 3a) Insertar Capa	
Acción del Actor	Respuesta del Sistema
	3a.1) Inserta la capa correspondiente a la Dependencia Policial.
	3a.2) Inserta el ícono que representa las Dependencias Policiales en su dirección correspondiente.
	3a.3) Muestra en el mapa el ícono de la dependencia

	policial obtenida.
	3a.4) Ir a la acción 6.
Poscondiciones	No aplica.

Prototipo de interfaz de usuario



El resto de las descripciones de casos de uso se encuentran en el Anexo 1.

2.6 Conclusiones

En el presente capítulo se muestra una visión de las características que debe cumplir el sistema. Se presenta el modelo del dominio, se enumeran los requerimientos funcionales y no funcionales, se identifican los actores y casos de usos del sistema. Se realizan los diagramas paquetes y de casos de usos del sistema respectivamente. Se identificó la información geográfica presente en SIGEPOL para facilitar la especificación de las nuevas funcionalidades a implementar en el SMW dando así cumplimiento al segundo objetivo específico del trabajo de diploma.

Capítulo 3. Análisis y Diseño del Sistema

3.1 Introducción

En el presente capítulo se plantea el modelo del análisis donde se obtiene una visión del sistema en el que solo interesa QUÉ el sistema debe hacer, por lo tanto se trabaja con los requerimientos funcionales definidos en el capítulo anterior. Se presenta también el diagrama de paquetes y el diagrama de clases correspondiente tanto para el análisis como para el diseño. Se detallan los cambios en la arquitectura del SMW. Se especifican los diagramas de clases del diseño los cuales constituyen un refinamiento del análisis que cuenta con los requerimientos no funcionales, es decir CÓMO el sistema cumple sus objetivos, este flujo de trabajo es de gran importancia pues permite que el sistema pueda ser implementado sin ambigüedades.

3.2 Análisis

Se obtiene el modelo de análisis, donde se identifican las clases del análisis, las cuales están asociadas al contexto del dominio del problema y describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición es de gran importancia en el diseño e implementación de la solución.

Un diagrama de clases del análisis constituye el artefacto en el que se representan los conceptos del dominio del problema, pues representa elementos del mundo real y no de la implementación.

Los diagramas de clases del análisis se realizaron por paquetes, facilitando el agrupamiento de los requisitos funcionales teniendo en cuenta su comportamiento y la información que utilizan los mismos. A continuación, se presenta el diagrama de paquetes del análisis.

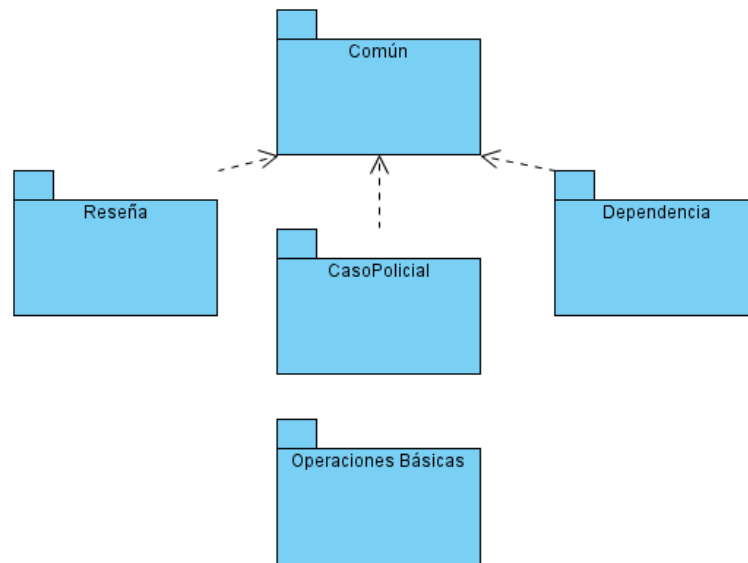


Fig. 3.1 Diagrama Paquetes del Análisis

3.2.1 Diagrama de Clases del Análisis Paquete Operaciones Básicas

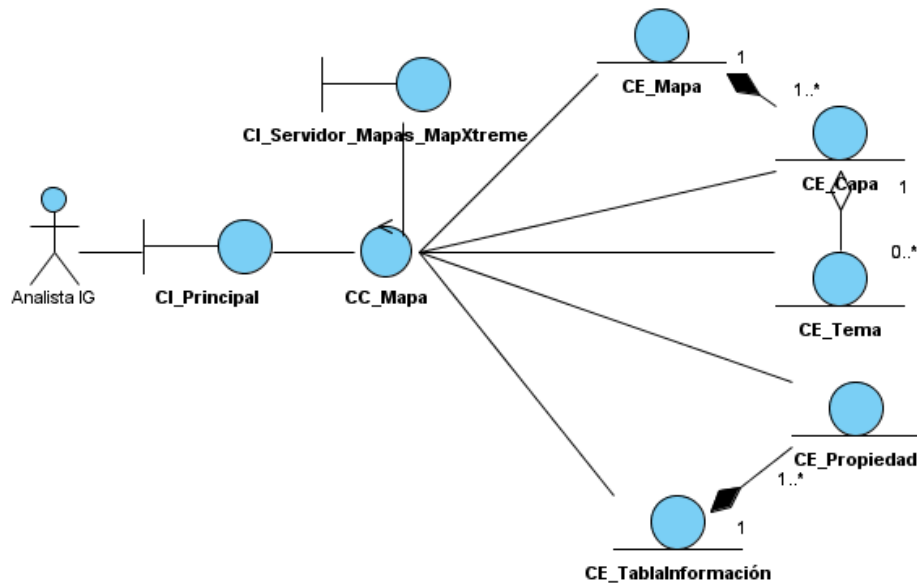


Fig. 3.2 Diagrama de Clases del Análisis Paquete Operaciones Básicas

El resto de los diagramas de clases del análisis se encuentran en el Anexo 2.

3.3 Arquitectura

La arquitectura de software es el diseño de más alto nivel de la estructura o esqueleto de un sistema. Es diseñada en base a requerimientos funcionales, no funcionales y de restricciones preestablecidas. Se tiene en cuenta para su elaboración el grado de flexibilidad y escalabilidad que el sistema debe tener; además de la interacción que pueda existir con otros sistemas. Se identifican los componentes más significativos y se especifica la comunicación entre ellos. La arquitectura constituye la base sólida para el correcto funcionamiento de una aplicación.

3.3.1 Propuesta de Arquitectura

Se propone a continuación el diseño modular de la arquitectura del SMW, mediante la especificación de módulos coherentes e independientes con funcionalidades bien definidas. Este diseño es más efectivo que el anterior, pues simplifica la complejidad de implementación permitiendo el desarrollo en paralelo de los módulos del sistema teniendo en cuenta el propósito de cada uno.

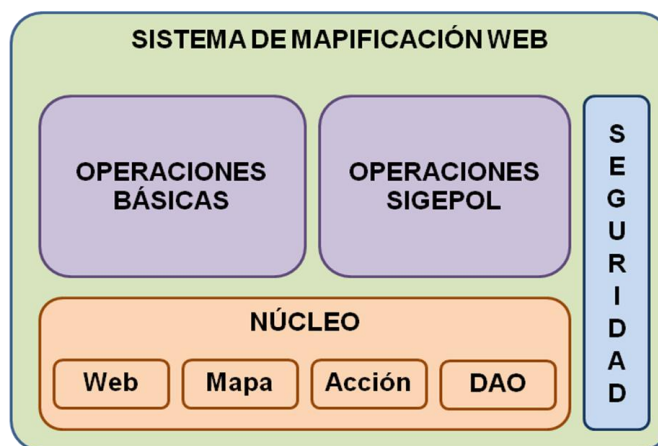


Fig. 3.3 Arquitectura del Sistema

Con la arquitectura presentada se garantiza un alto nivel de organización y extensibilidad del sistema, pues las funcionalidades se agrupan convenientemente en módulos que realizan una lógica de negocio

determinada. Permitiendo agregar o eliminar funcionalidades al módulo que corresponda en caso de que sea necesario.

La arquitectura se encuentra fragmentada en cuatro módulos que agrupan y gestionan una serie de funcionalidades de acuerdo al propósito de cada uno.

Módulo Núcleo

De vital importancia pues constituye el corazón del sistema pues brinda soporte a los módulos Operaciones Básicas y Operaciones SIGEPOL. Se compone por los paquetes **Web**, **Mapa**, **Acción** y **DAO**.

En el paquete **Web** se encuentra **MapaControlador**, controlador responsable de manejar las peticiones relacionadas con las acciones que pueden ser ejecutadas sobre los mapas. El paquete **Mapa** representa el comportamiento de un mapa real, pues encapsula el modelo de datos que es mostrado digitalmente en una vista al usuario ya que conserva la información correspondiente en un objeto de la clase **MapJ** del API del framework **MapInfo MapXtreme**.

El paquete **Acción** constituye la base de las acciones que pueden realizarse sobre mapas mediante la interfaz **IAccion**; está también la clase **FabricaAcciones** encargada de gestionar los pares acción-vista de las URL definidas en el contexto de la aplicación, permitiendo abstraer al **MapaControlador** de identificar quién debe atender qué acción según la petición recibida.

El paquete **DAO** contiene la clase **AbstractIbatisDAO** soportada por el framework **Ibatis** la cual permite la ejecución de procedimientos almacenados en aquellas clases que hereden de la misma.

Módulo Operaciones Básicas

Contiene las funcionalidades básicas de navegación sobre mapas que provee el sistema mediante acciones que se ejecutan en dependencia de la petición recibida, encapsulando la información con la que se trabaja. Dichas acciones se encuentran soportadas sobre el módulo Núcleo para su correcto desempeño.

Módulo Operaciones SIGEPOL

Engloba las funcionalidades relacionadas con el procesamiento y visualización de la información geográfica de SIGEPOL, estas acciones se encuentran organizadas en los paquetes Dependencia, Caso Policial, Reseña y Común de acuerdo a la información que utilizan.

Cuando se recibe una petición correspondiente al módulo Operaciones SIGEPOL la comunicación con el mismo se establece a través el módulo Núcleo mediante **MapaControlador**, el cual es el controlador responsable de manejar las peticiones relacionadas con la acciones de mapificación que pueden ser ejecutadas sobre los mapas. En este controlador la acción y la vista correspondiente a la petición recibida se obtienen de la clase **FabricaAcciones**. Luego se ejecuta dicha acción en el contexto del módulo Operaciones SIGEPOL obteniéndose entonces el modelo del mapa a mostrar en la vista al usuario.

Módulo Seguridad

Se encarga de controlar los aspectos relacionados con la seguridad de la aplicación mediante el control de la autenticación de usuario así como de gestionar las políticas de seguridad definidas. Este módulo se desarrolla utilizando el framework Spring Security, donde se filtran las llamadas a los controladores de la aplicación, verificando que el usuario esté autenticado y que posea el rol que le permite hacer uso de las funcionalidades. Éste módulo se representa en la arquitectura de manera vertical debido a que la seguridad es gestionada en todo el sistema.

La arquitectura base de cada módulo de la aplicación es en Capas, muy similar a la primera versión de la arquitectura del SMW, donde las capas definidas son **Dominio**, **Presentación**, **Negocio** y **Acceso A Datos** como se muestra en la figura a continuación.

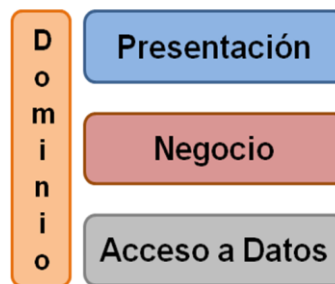


Fig. 3.4 Arquitectura Módulos

3.4 Diseño

Cómo propósito del diseño se plantea la comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos y tecnologías a utilizar. Se crea una entrada apropiada y un punto de partida para las

actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases. Se descomponen los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo.

3.4.1 Diagramas de Interacción

Existen dos tipos de diagramas de interacción en UML, colaboración y secuencia. Estos son utilizados para modelar los aspectos dinámicos del sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre sí.

En un diagrama de secuencia se destaca el orden temporal de los mensajes; mientras que un diagrama de colaboración es de mayor importancia la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de secuencia para flujo principal y flujos alternativos correspondientes a cada realización de caso de uso aparecen en el Anexo 3.

3.4.2 Diagrama de Clases del Diseño

El diagrama general de clases del diseño representa la interacción entre las clases de cada capa por las que está integrada la aplicación. Se representa el color de las clases de cada capa como se especifica en la figura 3.4. Las clases de color naranja son aquellas que pertenecen al módulo Núcleo. Por último las clases representadas con el color verde indican las clases que pertenecen a los frameworks utilizados para una mayor claridad de los diagramas.

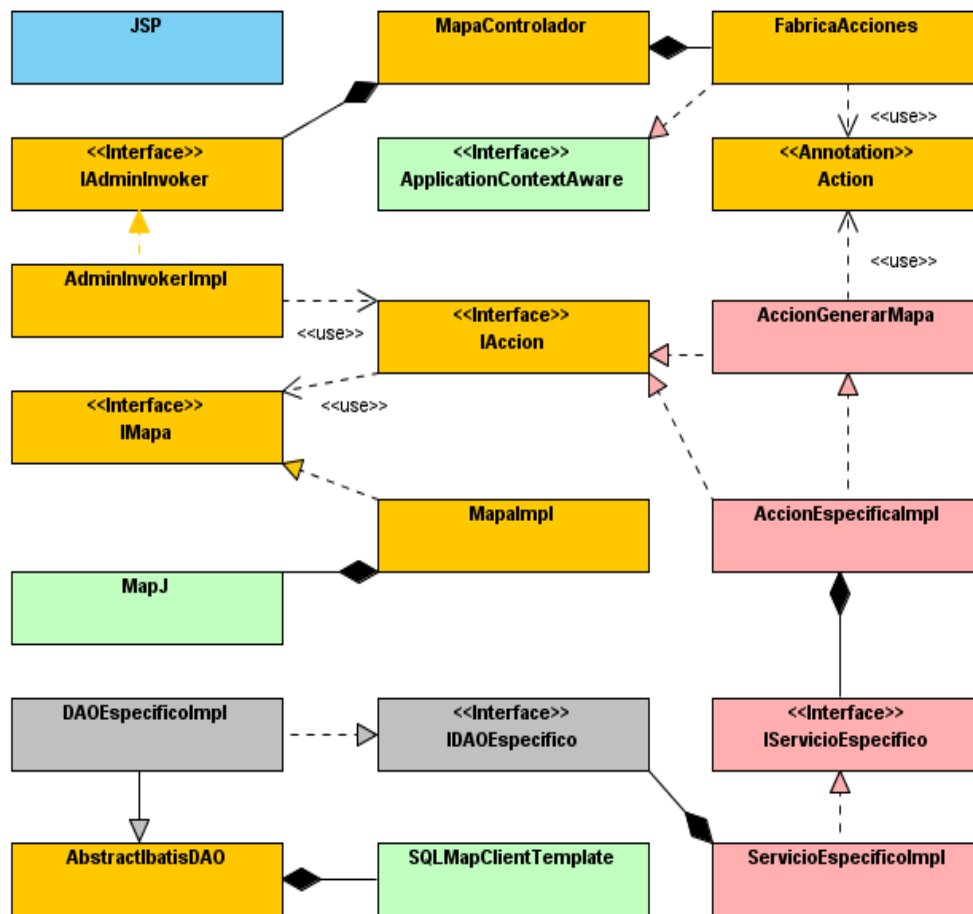


Fig. 3.5 Diagrama de Clases del Diseño General

3.4.3 Diagrama de Paquetes del Diseño

Los diagramas de paquetes son de gran utilidad para organizar el sistema en elementos más pequeños y fáciles de interpretar. Estos revelan cómo es la estructura de un sistema y cómo se encuentra dividido por conjuntos lógicos mostrando las relaciones que existen entre los mismos. Un paquete de diseño proporciona una colección de clases, realizaciones de casos de uso, diagramas, relaciones y otros paquetes.

Para la confección del diagrama de paquetes que se muestra a continuación se decidió agrupar las clases con respecto al tipo de funcionalidades que ejecutan. Facilitando el entendimiento de la

organización de las clases y de la arquitectura del sistema. Se muestra además las colaboraciones entre paquetes pues se visualiza la relación establecida entre ellos; por ejemplo, el paquete A depende de B cuando A invoca las funcionalidades contenidas en B.

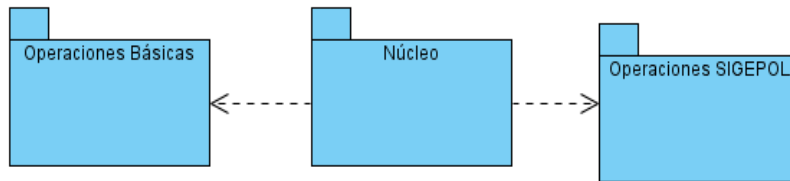


Fig. 3.6 Diagrama de Paquetes del Diseño General

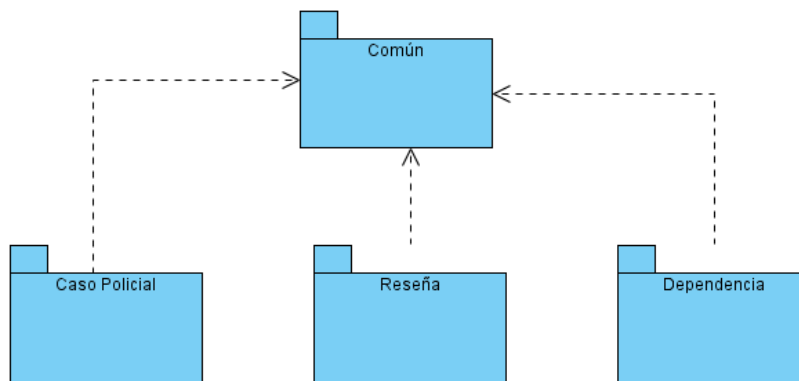


Fig. 3.7 Diagrama de Paquetes del Diseño Operaciones SIGEPOL

3.4.4 Diagrama de Clases del Diseño Paquete Núcleo

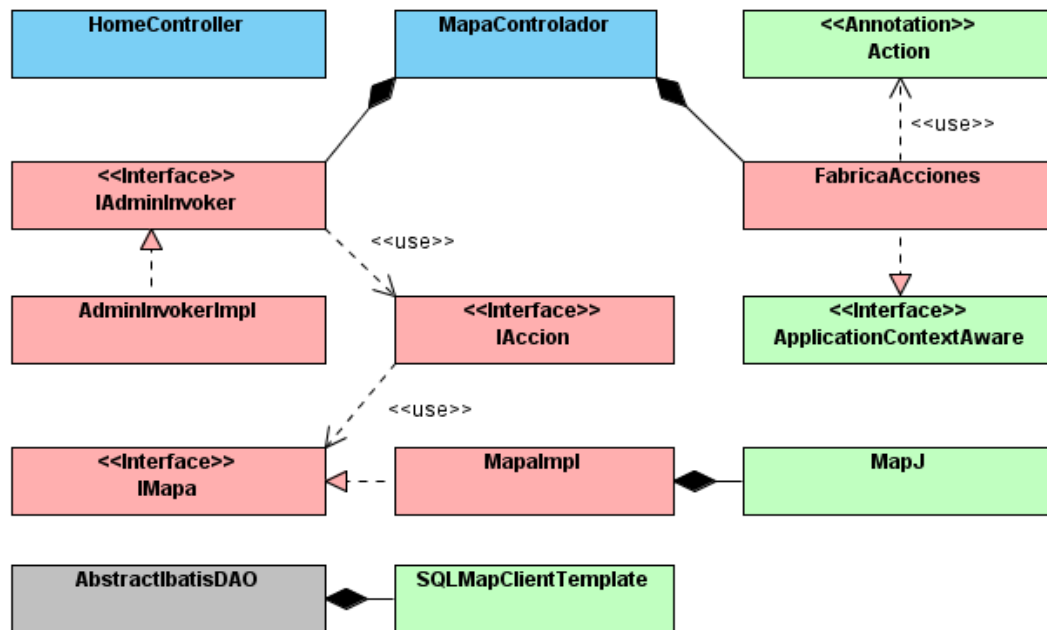


Fig. 3.8 Diagrama de Clases del Diseño Paquete Núcleo

Los demás diagramas de clases del diseño se encuentran ubicados en el Anexo 4.

3.5 Modelo Lógico de Datos

Este modelo provee una vista de las entidades lógicas de datos así como sus relaciones independientemente de la plataforma de base de datos a utilizar. Seguidamente se muestra el diagrama de clases persistentes de la aplicación, el cual modela las clases mediante las que se obtiene la información.

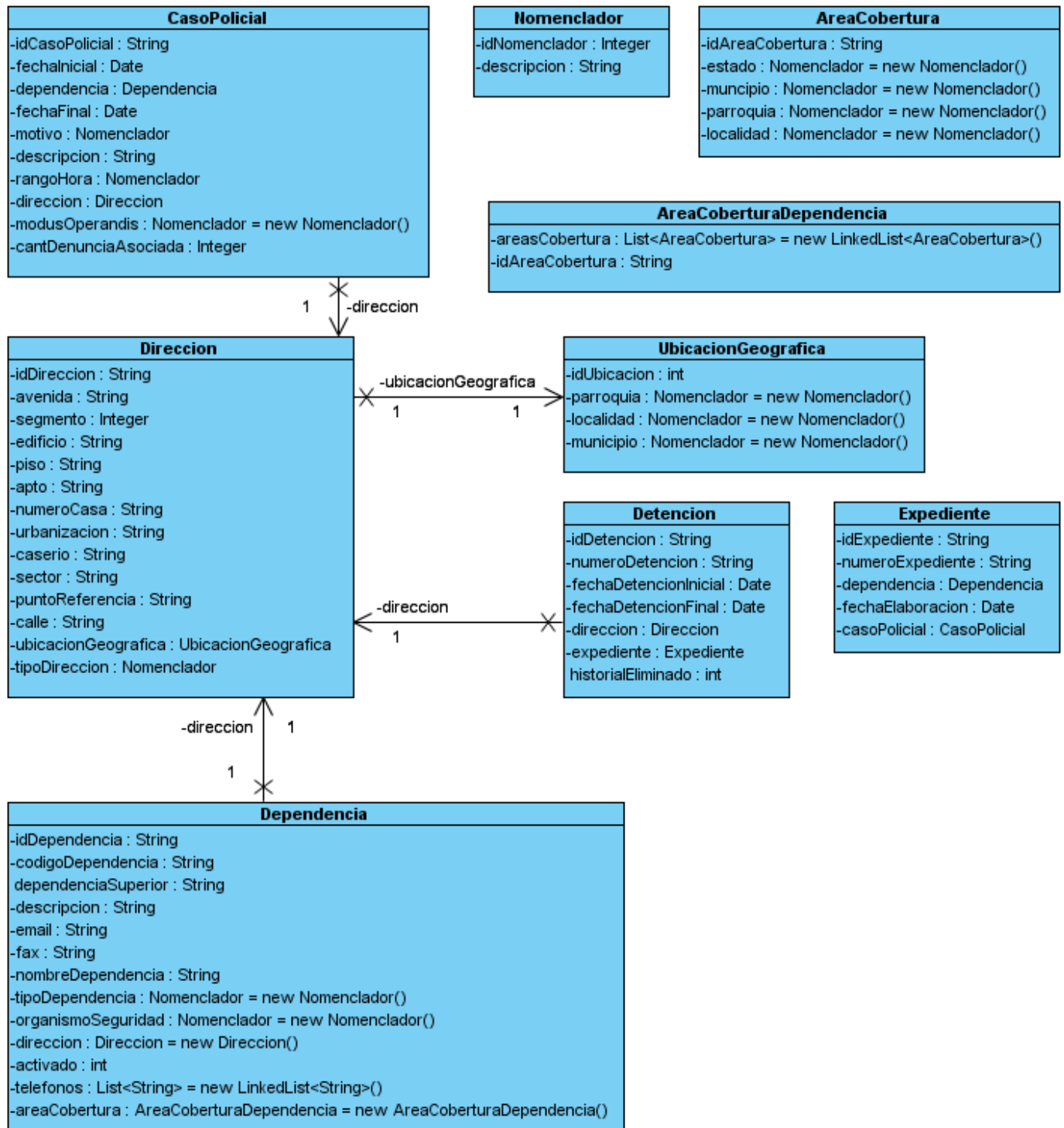


Fig. 3.9 Diagrama de Clases Persistentes

3.6 Modelo Físico de datos

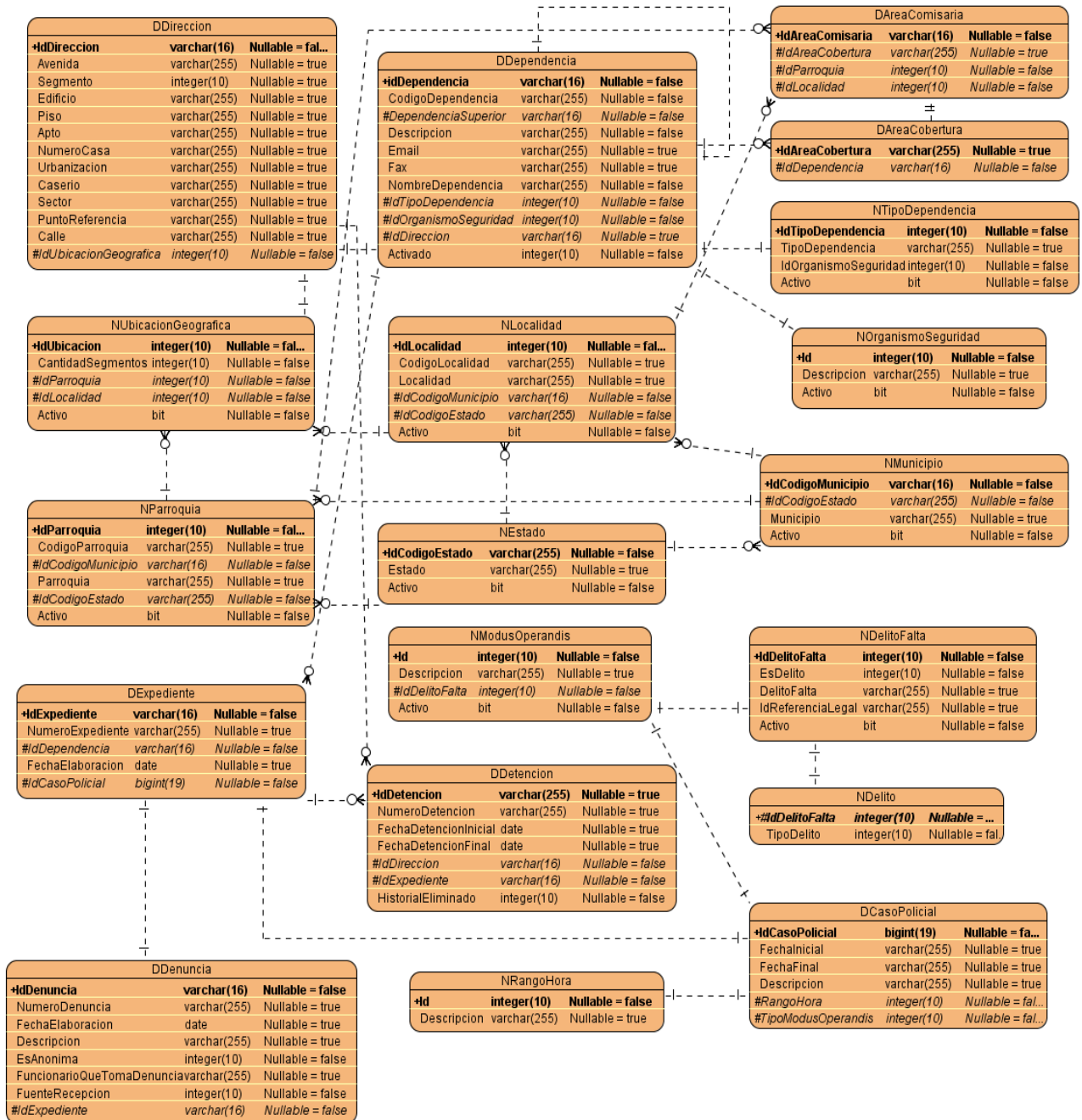


Fig. 3.10 Diagrama Entidad – Relación

El diagrama Entidad – Relación refleja la estructura natural de los datos. Representa entidades y las conexiones que existen entre estas, es decir, las tablas de la base de datos que fueron utilizadas y sus relaciones.

3.7 Conclusiones

El presente capítulo proporciona un punto de partida para la implementación del sistema, logrando un mejor entendimiento de los aspectos relacionados con los requisitos definidos en el capítulo anterior. Se definieron las clases del análisis para modelar los conceptos del mundo real y se especificaron los diagramas de clase del diseño por paquetes. Se detallaron los aspectos generales de la arquitectura cumpliendo el tercer objetivo específico del trabajo de diploma.

Capítulo 4. Implementación

4.1 Introducción

En el presente capítulo se encuentra representado el modelo de implementación, el cual constituye la organización y dependencia entre nodos físicos en los que funcionará la aplicación, donde los elementos especificados en el modelo del diseño son traducidos en términos de componentes elaborándose los diagramas de componentes. Se exponen los principales diagramas de componentes que conforman el SMW, en la cual por cada componente se encuentra una breve descripción donde consta su contenido y propósito. También se muestra el diagrama de despliegue de la aplicación.

4.2 Modelo de Implementación

El modelo de implementación está conformado por el diagrama de componentes y el diagrama de despliegue. Se describen los componentes a construir así como su organización y dependencia entre nodos físicos en los que funcionará el sistema.

4.2.1 Diagrama de Despliegue

Este diagrama se utiliza para capturar los elementos de configuración del procesamiento de la información y las conexiones que existen entre estos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. El diagrama está conformado por los siguientes elementos:

- ✓ **Nodos:** Elementos de procesamiento con al menos un procesador, memoria y posiblemente otros dispositivos.
- ✓ **Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- ✓ **Conectores:** Expresa el tipo de conector o protocolo utilizado entre los elementos del diagrama.

A continuación, se presenta el diagrama de despliegue definido.

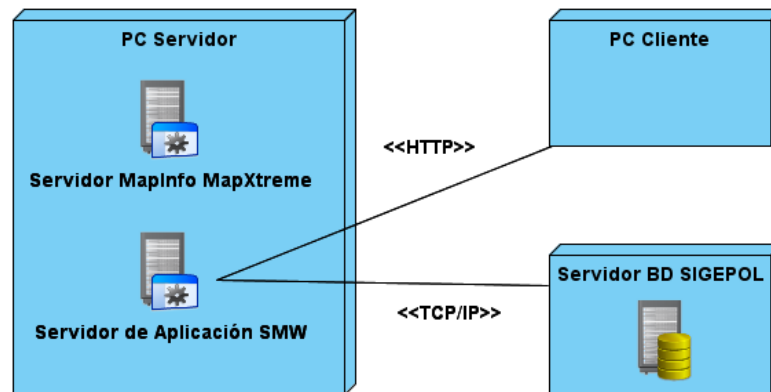


Fig. 4.1 Diagrama de Despliegue

4.2.2 Diagrama de Componentes

Los diagramas de componentes se utilizan para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los mismos. Su principal objetivo es mostrar la estructura de alto nivel del modelo de implementación, mostrando la organización y las dependencias lógicas entre un conjunto de componentes de software que pueden ser por ejemplo código fuente, librerías, ficheros binarios o ejecutables. Seguidamente se muestra el diagrama de componentes correspondiente a la aplicación.

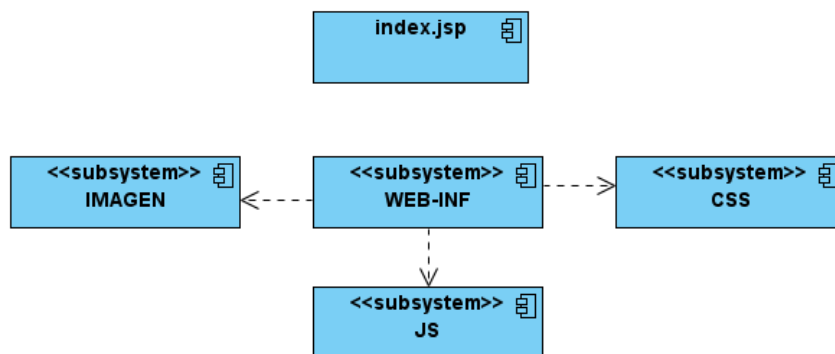


Fig. 4.2 Diagrama de Componentes

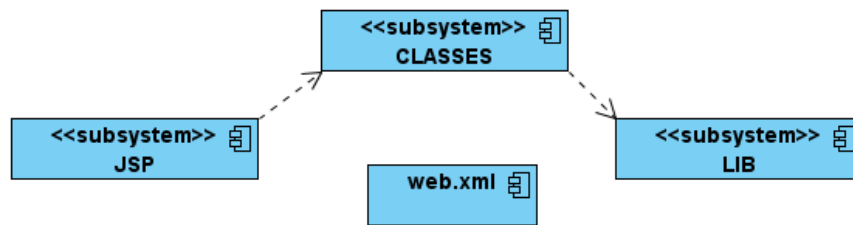


Fig. 4.3 Diagrama de Componentes Subsistema WEB-INF

4.2.2.1 Descripción del Subsistema de Implementación Classes

Subsistema CLASSES

Propósito:

Contenedor físico del compilado de cada una de las clases de la aplicación, ficheros de configuración y archivos con extensión JAVA y XML.

Contenido:

NUCLEO.accesodatos

- ✓ AbstractIbatisDAO.class

NUCLEO.acciones

- ✓ FabricaAcciones.class
- ✓ IAccion.class

NUCLEO.admin

- ✓ IAdminInvoker.class
- ✓ AdminInvokerImpl.class

NUCLEO.mapa

- ✓ IMapa.class
- ✓ MapaImpl.class

NUCLEO.stereotype

- ✓ Action.class

NUCLEO.web

- ✓ HomeController.class
- ✓ MapaContrador.class

NUCLEO.servicio

- ✓ IServicioMapa.class

NUCLEO.servicio.imp

- ✓ ServicioMapaImpl.class

CONFIGURACION

- ✓ app-config.xml
- ✓ controller-config.xml
- ✓ ibatis-config.xml
- ✓ mapa-config.xml
- ✓ seguridad-config.xml
- ✓ sql-map-config.xml

BASICAS.acciones

- ✓ AccionAcercarMapa.class
- ✓ AccionAlejarMapa.class
- ✓ AccionCentrarMapa.class
- ✓ AccionCentrarOverview.class
- ✓ AccionDeseleccionarMapa.class
- ✓ AccionEliminarElemento.class
- ✓ AccionGenerarMapa.class
- ✓ AccionGenerarMapaTematico.class

- ✓ AccionInformacionRegion.class
- ✓ AccionInsertarLinea.class
- ✓ AccionInsertarPunto.class
- ✓ AccionMostrarTodasCapas.java
- ✓ AccionSeleccionarMapa.class
- ✓ AccionSeleccionRectangular.class
- ✓ AccionVisibilidadCapas.class

BASICAS.dominio.mapa

- ✓ Capa.class
- ✓ Mapa.class
- ✓ Propiedad.class
- ✓ TablaInformacion.class
- ✓ Tema.class
- ✓ TipoEvento.class

SIGEPOL.accesodatos

- ✓ ICasoPolicialDao.class
- ✓ IDetencionDao.class
- ✓ IDependenciaDao.class
- ✓ INomencladorDao.class

SIGEPOL.accesodatos.impl

- ✓ CasoPolicialDaoImpl.class
- ✓ DetencionDaoImpl.class
- ✓ NomencladorDaoImpl.class
- ✓ DependenciaDaoImpl.class

SIGEPOL.accesodatos.map

- ✓ CasoPolicial.xml

- ✓ Dependencia.xml
- ✓ Nomenclador.xml
- ✓ NomencladorDireccion.xml
- ✓ Resenna.xml

SIGEPOL.acciones.casoPolicial

- ✓ AccionMostrarCasosPoliciales.class
- ✓ AccionMostrarDetalleCasoPolicial.class

SIGEPOL.acciones.dependencia

- ✓ AccionMostrarAreaTodasDependencias.class
- ✓ AccionMostrarDependencias.class
- ✓ AccionMostrarDetalleDependencias.class
- ✓ AccionMostrarTodasDependencias.class

SIGEPOL.acciones.resenna

- ✓ AccionMostrarDetenciones.class
- ✓ AccionMostrarDetalleDetencion.class

SIGEPOL.dominio.casoPolicial

- ✓ CasoPolicial.class

SIGEPOL.dominio.comun

- ✓ AreaCobertura.class
- ✓ Direccion.class
- ✓ IListable.class
- ✓ Nomenclador.class
- ✓ NomencladorDireccion.class
- ✓ NomencladorDireccion.class
- ✓ TablaPaginacion.class

- ✓ UbicacionGeografica.class

SIGEPOL.dominio.dependencia

- ✓ AreaCoberturaDependencia.class
- ✓ Dependencia.class

SIGEPOL.dominio.resenna

- ✓ Detencion.class
- ✓ Expediente.class

SIGEPOL.servicio

- ✓ IServicioCasoPolicial.class
- ✓ IServicioControlador.class
- ✓ IServicioDependencia.class
- ✓ IServicioDetencion.class
- ✓ IServicioNomenclador.class

SIGEPOL.servicio.impl

- ✓ ServicioCasoPolicialImpl.class
- ✓ ServicioControladorImpl.class
- ✓ ServicioDependenciaImpl.class
- ✓ ServicioDetencionImpl.class
- ✓ ServicioNomencladorImpl.class

SIGEPOL.web.controlador

- ✓ CasoPolicialControlador.class
- ✓ DependenciaControlador.class
- ✓ DetencionControlador.class

SIGEPOL.web.nomenclador

- ✓ NomencladorDelito.class
- ✓ NomencladorDependenciaControlador.class
- ✓ NomencladorDireccionControlador.class

El resto de las descripciones de los componentes y subsistemas de implementación aparecen en el Anexo 5.

4.3 Conclusiones

Este capítulo es de gran importancia pues muestra la distribución de los servidores involucrados para el correcto despliegue del SMW. También se representa la estructura física del sistema donde se identifican componentes y subsistemas de implementación además de especificar el propósito y contenido de cada uno y las relaciones que existen entre ellos. De esta manera se da cumplimiento al cuarto objetivo específico del trabajo de diploma.

Capítulo 5. Estudio de Factibilidad

5.1 Introducción

El estudio de factibilidad es de vital importancia en el proceso de desarrollo de cualquier proyecto, pues posibilita determinar que tan productivo puede llegar a ser un proyecto de software, así como también permite estimar el tiempo de demora para su desarrollo. En este capítulo se realiza un análisis de los costos, beneficios y el esfuerzo evaluado del proyecto.

5.2 Estimación de Puntos por Caso de Uso

Método de estimación del esfuerzo de desarrollo de un producto de software teniendo en cuenta los casos de uso, se realiza mediante la asignación de pesos a un cierto número de factores que lo afectan en complejidad técnica y ambiente.

5.2.1 Cálculo de Puntos de Caso de Usos sin Ajustar

Para calcular los puntos de casos de uso sin ajustar se aplica la siguiente fórmula:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Para determinar el Factor de Peso de los Actores sin ajustar (UAW):

Para determinar el Factor de Peso de los Actores sin ajustar (UAW):

Tipo	Descripción	Peso	Cant*Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante un API.	1	1*1

Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	1*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	0*3
Total		3	

Para determinar el Factor de Peso de los Casos de Uso sin ajustar (UUCW):

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15

Aplicando lo anterior a nuestro sistema se obtiene como resultado el Anexo 6.1.

Existen 11 casos de uso, 6 con complejidad simple, 4 con complejidad media y 1 complejos. Calculándose entonces el factor de peso de los casos de uso sin ajustar:

$$UUCW = 6*5 + 4*10 + 1*15 = 85.$$

$$UUCP = 3 + 85 = 88$$

5.2.2 Cálculo de Puntos de Casos de Uso Ajustados

Una vez que se tienen los puntos de casos de uso sin ajustar, se debe ajustar este valor mediante la siguiente fórmula:

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El coeficiente TCF se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del software. A cada uno de los factores se le asigna un valor de 0 a 5 de acuerdo a su relevancia. La fórmula utilizada es la siguiente:

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor Asignado}_i)$$

Significado de los valores

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

La tabla correspondiente al cálculo del TCF se encuentra en el Anexo 6.2.

Se obtiene entonces como factor de complejidad técnica:

$$TCF = 0.6 + 0.01 * 55 = 1.15$$

Para el cálculo del factor de ambiente se consideran las habilidades, entrenamientos y experiencias del grupo de desarrollo, donde estas tienen un gran impacto en las estimaciones de tiempo. La fórmula utilizada es la siguiente:

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i * \text{Valor Asignado}_i)$$

Factor	Descripción	Peso	Valor	Σ (Peso_i * Valori)
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	3	1.5

E3	Experiencia en orientación a objetos	1	5	5
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	5	10
E7	Personal part-time	-1	2	-2
E8	Dificultad del lenguaje de programación	-1	4	-4
Total				22

Constituyendo el factor de ambiente:

$$EF = 1.4 - 0.03 * 22 = 0.74$$

Entonces:

$$UCP = UUCP * TCF * EF$$

$$UCP = 88 * 1.15 * 0.74 = 74.888$$

5.2.3 Cálculo del Esfuerzo

El esfuerzo en Horas-Hombre se calcula mediante la fórmula:

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Cálculo de CF:

CF = 20 horas-hombre (si Total EF \leq 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Total EF = Cant EF < 3 (entre E1 – E6) + Cant EF > 3 (entre E7, E8)

Como:

$$\text{Total EF} = 2 + 0 = 2$$

$$\text{CF} = 20 \text{ horas-hombre (porque Total EF} \leq 2)$$

Luego:

$$E = 74.888 * 20 \text{ horas-hombre}$$

$$E = 1497.76 \text{ horas-hombre}$$

5.2.4 Distribución del Esfuerzo

El Esfuerzo Total sería 1497.532 horas-hombre, teniendo en cuenta el Anexo 6.3 y las condiciones que presentan los desarrolladores de este software, que un mes tiene 176 horas laborables, pues se trabajan 8 horas diarias 22 días al mes, entonces el Esfuerzo Total sería 8.50 mes-hombre.

5.2.5 Costo del Proyecto

Para obtener el costo total del proyecto se emplea la siguiente fórmula:

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH}$$

Donde:

CH: Cantidad de hombres.

CHM: Costo Hombre – Mes.

ET: Esfuerzo Total.

Si la cantidad de hombres es 2 y se tiene un salario promedio mensual igual a \$100.00.

Entonces:

$$\text{CHM} = \text{CH} * \text{Salario Promedio} = 2 * 100 = 200.00$$

Luego:

$$\text{Costo} = \text{CHM} * \text{ET} / \text{CH} = 200.00 * 8.50 / 2 = \$ 850.00$$

5.2.6 Cálculo del Tiempo de Desarrollo del Proyecto

Tiempo = ET / CH

Tiempo = 8.50 / 2 = 4.25 ≈ 5 meses

5.3 Beneficios Tangibles e Intangibles

Debido a las nuevas funcionalidades incorporadas al SMW se obtiene como resultado el Sistema de Información Geográfica Policial (SIGPOL); destinada a las dependencias policiales de Venezuela con el propósito de procesar y visualizar en un mapa la información geográfica que maneja SIGEPOL. El principal beneficio que ofrece el sistema es la posibilidad de realizar estudios estadísticos policiales a partir de la información que brinda la aplicación, garantizando una mejor toma de decisiones en dependencia de la situación delictiva apreciada.

Los beneficios intangibles son aquellos que favorecen el funcionamiento o eficiencia de una organización o entidad. A continuación, los beneficios intangibles que depara el sistema:

Mejor apreciación de la información geográfica de SIGEPOL, pues en dicho sistema esta información es mostrada generalmente a través de tablas dificultando la comprensión de los datos; mientras que con el uso de SIGPOL resulta más fácil realizar una valoración acerca del índice de criminalidad correspondiente a un área de cobertura de una dependencia policial por ejemplo, ya que esta información representada en un mapa favorece en gran medida su entendimiento.

Acceso a la aplicación en tiempo real las 24 horas, asegurando la disponibilidad de los servicios que brinda SIGPOL y que los mismos puedan ser utilizados cuando sea necesario.

5.4 Conclusiones

En el presente capítulo se realizó la estimación por Puntos de Casos de Uso, la cual permitió valorar el esfuerzo del proyecto teniendo en cuenta factores como la habilidad, experiencia y entrenamiento que presenta el grupo de desarrollo. Se estimó el costo y el tiempo de desarrollo del proyecto, así como se realizó un análisis de los beneficios tangibles e intangibles que provee el sistema, permitiendo conocer que tan factible resulta el desarrollo del proyecto.

Conclusiones Generales

Se ha dado solución al problema científico planteado, con el cumplimiento a las tareas de investigación propuestas, pues se estudiaron las soluciones anteriores para obtener un punto de partida. Se identificó la información geográficamente referenciada de SIGEPOL a visualizar en el mapa y se añadieron las nuevas funcionalidades al SMW permitiendo el procesamiento de dicha información.

Se realizó un estudio de las herramientas y tecnologías empleadas en el desarrollo de la aplicación. Para el proceso de desarrollo de software se empleó la metodología RUP, donde se realizaron los siguientes flujos de trabajo: Modelamiento del Negocio, Requerimientos, Análisis y Diseño e Implementación a largo de todas sus fases. Estos ciclos de desarrollo contribuyeron a un eficiente desenvolvimiento del perfeccionamiento de la aplicación y se demostró que el presente trabajo de diploma es viable mediante el estudio de factibilidad efectuado.

Se da cumplimiento al objetivo general planteado del trabajo de diploma mediante la integración de SIGEPOL y el SMW. Pues SIGPOL incorpora además de las operaciones básicas que ofrece el SMW, la búsqueda de elementos por criterios establecidos que posibilitan la visualización de dependencias policiales y sus respectivas áreas de responsabilidad. Permite representar la ubicación geográfica de detenciones y hechos delictivos, así como posibilita consultar información detallada de los elementos mostrados en el mapa.

El producto alcanzado en este trabajo de diploma trae como ventaja un mejor desenvolvimiento y eficiencia en el cumplimiento de las actividades que se realizan en las dependencias policiales de Venezuela. El sistema posibilita la visualización de eventos relevantes en un mapa, haciendo más sencilla la interpretación de esta información de una manera que no pudiera ser apreciada en SIGEPOL. Por ejemplo, se puede identificar zonas calientes con mayor facilidad, es decir, se garantiza el reconocimiento de lugares donde existe un número considerable de hechos delictivos, permitiendo que los esfuerzos policiales puedan ser concentrados hacia esa área. También figura el descubrimiento y análisis de patrones de comportamiento así como tendencias a partir de los datos mostrados.

Recomendaciones

- ✓ Garantizar la veracidad de las direcciones que se registran en la base de datos de SIGEPOL, pues así se efectuará una correcta ubicación de los eventos en el mapa, es decir, la información mostrada por la aplicación será más confiable y consistente.
- ✓ Realizar las pruebas de software para evaluar y garantizar la calidad del sistema.

Referencias Bibliográficas

1. **González Tejada, Ing. Reynier.** SISTEMA DE MAPIFICACIÓN WEB. Universidad de las Ciencias Informáticas : Tesis, 2009.
2. **Oficina de Tecnologías de Información de MPPRIJ.** Ministerio del Poder Popular para Relaciones Interiores y de Justicia. [En línea] [Citado el: 11 de Febrero de 2010.] <http://www.mpprij.gob.ve>.
3. **Consejo General de Policía.** Policía Nacional Bolivariana. [En línea] [Citado el: 15 de diciembre de 2009.] <http://www.policianacional.gob.ve/index.php/la-nueva-policia-nacional/vision-policia-nacional>.
4. **Góngora Ramos, Ing. Yunier y Álvarez Pérez, Ing. Oigres.** SISTEMA DE GESTIÓN POLICIAL. MÓDULO DE OPERATIVOS POLICIALES. Universidad de las Ciencias Informáticas : Tesis, 2008.
5. **The Regents of the University of California .** National Center for Geographic Information and Analysis. [En línea] [Citado el: 14 de Enero de 2010.] <http://www.ncgia.ucsb.edu>.
6. **Instituto Geográfico Militar.** Instituto Geográfico Militar. [En línea] [Citado el: 11 de Febrero de 2010.] http://www.igm.gov.ec/cms/index.php?option=com_content&task=view&id=26&Itemid=71.
7. **AEROTERRA.** AEROTERRA. Soluciones geográficas. [En línea] [Citado el: 17 de Febrero de 2010.] <http://www.aeroterra.com/so-delito.htm>.
8. **Pitney Bowes Software Inc.** Pitney Bowes Business Insight. [En línea] [Citado el: 2 de Febrero de 2010.] <http://www.pbinsight.com/>.
9. **Java.** Sitio oficial Java. [En línea] [Citado el: 4 de Febrero de 2010.] <http://www.java.com>.
10. **Román Zamitiz, Ing. Carlos Alberto.** Programación con Java. [En línea] [Citado el: 4 de Febrero de 2010.] http://profesores.fi-b.unam.mx/carlos/java/java_basico1_2.html.
11. **Ext JS, Inc.** Ext JS. [En línea] [Citado el: 3 de Febrero de 2010.] <http://www.extjs.com/>.
12. **Rod Johnson, Juergen Hoeller, Keith Donald, Colin Sampaleanu, Rob.** *Spring Framework Reference 3.0.* 2009.
13. **Apache Software Foundation.** IBATIS. [En línea] [Citado el: 3 de Marzo de 2010.] <http://ibatis.apache.org/about.html>.
14. **Jacobson, I. y Booch, G. y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de software.* 2000.

15. **The Eclipse Foundation.** Eclipsecon. [En línea] [Citado el: 7 de Febrero de 2010.]
<http://www.eclipse.org/org/>.

16. **Diego, Juan.** Herramientas CASE. [En línea] 24 de Mayo de 2008. [Citado el: 7 de Febrero de 2010.]
<http://aquiredactoyo.blogspot.com/>.

17. **Visual Paradigm International Ltd.** Visual Paradigm. [En línea] [Citado el: 7 de Febrero de 2010.]
<http://www.visual-paradigm.com/>.

Bibliografía

Alarcón Ferrá, Ailin y Arias Arias, Michel. 2007. SISTEMA DE GESTIÓN DE EMERGENCIAS DE SEGURIDAD CIUDADANA (171). Módulo de Mapificación sobre Web. Universidad de las Ciencias Informáticas : Tesis, 2007.

Bello Ross, Ing. Elieser. 2008. SISTEMA DE GESTIÓN POLICIAL. DISEÑO DE BASE DE DATOS. Universidad de las Ciencias Informáticas : Tesis, 2008.

González Tejada, Ing. Reynier. 2009. SISTEMA DE MAPIFICACIÓN WEB. Universidad de las Ciencias Informáticas : Tesis, 2009.

MapData S.A. Mapdata. [En línea] [Citado el: 3 de Febrero de 2010.] http://www.mapdata.com.mx/productos_mapinfo02_Java.html.

MapInfo Corporation. 2006. *MapXtreme Java Edition, Developer Guide*. 2006.

Microsoft Corporation. 2009. MSDN Magazine. [En línea] Agosto de 2009. [Citado el: 24 de Febrero de 2010.] <http://msdn.microsoft.com/es-es/magazine/ee236415.aspx>.

R&A Marketing. Mapping Interactivo. [En línea] [Citado el: 11 de Febrero de 2010.] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=797.

Rigau i Claramunt, German. Servidor Adimen. [En línea] [Citado el: 24 de Febrero de 2010.] <http://adimen.si.ehu.es/~rigau/teaching/EHU/ISHAS/Curs2007-2008/Apunts/IS.4.pdf>.

Sánchez Téllez, Eddy. 2008. *Especificación de la Arquitectura Base del Sistema de Gestión de Emergencia y Seguridad Ciudadana 171*. Universidad de las Ciencias Informáticas : s.n., 2008.

SpringSource. Spring Source Community. [En línea] <http://www.springsource.org/>.

Glosario de Términos

Applet: Componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El applet debe ejecutarse en un contenedor, que lo proporciona un programa anfitrión pues no puede ejecutarse de manera independiente, ofrece información gráfica y a veces interactúa con el usuario.

Cross-Browser: Clasificación de aquellos sitios web, que han sido desarrollados para operar sobre varios navegadores.

FDD: Metodología de desarrollo de software ágil y escalable para proyectos con tiempo de desarrollo relativamente cortos. Orientado a funcionalidades además de que no es necesaria la presencia del cliente en el equipo.

HashMap: Colección de objetos, que carecen de orden alguno entre sí, la estructura atómica del mismo la constituye un par clave – objeto, de manera que se puede acceder al objeto por la clave especificada. El nombre Hash, hace referencia a una técnica de organización de archivos llamada hashing o "dispersión" en el cual se almacena el registro en una dirección del archivo que es generada por una función que se aplica sobre la llave del registro.

MapJ: Clase del framework MapXtreme encargada de contener la información del mapa digital que se muestra al usuario.

Módulo: Conjunto de funcionalidades acopladas convenientemente para realizar una lógica de negocio determinada. Es totalmente autónomo por lo que define claramente sus fronteras administrativas de seguridad, auditoría, mecanismos de interacción, configuración y aislamiento.

Modus Operandi: Su definición se corresponde con la manera especial de actuar para conseguir determinado fin. Aplicado a SIGEPOL es una pequeña descripción que se registra del incidente ocurrido correspondiente al hecho delictivo en cuestión.

Plug-in: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Subversion: Software para el control de versiones de un sistema, es decir, mantiene el registro de todo el trabajo y los cambios realizados sobre los ficheros que forman un proyecto. Posibilita que distintos desarrolladores (potencialmente situados a gran distancia) trabajen sobre la misma aplicación.

Widget: En el contexto de la programación de aplicaciones visuales se encuentra ligado a la de interfaz gráfica de usuario (GUI)¹⁶, es un componente gráfico, o control, con el cual el usuario interactúa y los programadores los utilizan para construir interfaces gráficas de usuario.

TCP/IP: Transmission Control Protocol/Internet Protocol por sus siglas en inglés es un conjunto de protocolos desarrollados para permitir a un computador o usuario compartir recursos de comunicación a través de una red.

Tiff: La denominación en inglés Tagged Image File Format es un formato de archivo de imágenes con etiquetas. Esto se debe a que los ficheros *.tiff contienen, además de los datos de la imagen propiamente dicha, "etiquetas" en las que se archiva información sobre las características de la imagen, que sirve para su tratamiento posterior.

XP: Metodología ágil que reduce la complejidad del software mediante un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

¹⁶ Graphic User Interface.