



Universidad de la Ciencias Informáticas

Facultad 2

Módulo Vigilancia y Patrullaje de la Personalización y Ampliación del SIIPOL para el Cuerpo de Policía Nacional Bolivariana

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Eleandro Emilio Pérez Acuña

Tutor: Ing. Carlos Molina Villalobos

Cotutores:

Ing. Alejandro Morales Torres

Ing. Yadira Marín González

Ciudad de la Habana, Junio de 2011

*"Es justamente la posibilidad de realizar un sueño lo
que hace que la vida sea interesante"*

Paulo Coelho

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al centro ISEC de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 22 días del mes de junio del año 2011

Eleandro Emilio Pérez Acuña

Ing. Carlos Molina Villalobos

Firma del Autor

Firma del Tutor

Ing. Alejandro Morales Torres

Ing. Yadira Marín González

Firma del Tutor

Firma del Tutor

Agradecimientos

A mima y pipa, por tanto y tanto amor.

A la Jelen, Ramón e Isabelita por desbordar la alegría del hogar.

A mi abuela, por su cariño y comprensión.

A toda mi familia y amigos, por su confianza sin límites.

A Elvira y familia, por recibirme en su hogar como uno más.

A Yadira y Alejandro porque sin su ayuda no me pudiera graduar.

A Katia, la mejor compañera de tesis.

A Liu, Yaro, Nairo, Pidio, Yudenia, Ivelis, Nayansi, Anagel, Raudel, siempre seremos 101.

A mi gente de la comunidad, por tantos buenos momentos.

A las Hermanas Lauritas y los padres Raúl, Carlos Juan e Iván.

A Antonia, Célida, Lily, Tona, Niurka, Nila, el Chino, Elvita.

A las amistades de estos años en la UCI, por tantos buenos momentos.

A la gente del proyecto, se extrañarán los Coffe Break.

A mi tutor por hacer de cada tarea un reto.

Dedicatoria

A Dios,

Diseñador y Arquitecto de la obra de mi vida.

Resumen

En la actualidad la falta de seguridad ciudadana es una situación que afecta a la población venezolana. Durante los últimos años los temas de convivencia y seguridad comenzaron a ser vistos por los ciudadanos como importantes problemas que deben ser solucionados por los mandatarios nacionales y locales. Con el propósito de responder a estas necesidades en abril del año 2008 fue publicado el Decreto N° 5.895 con Rango, Valor y Fuerza de Ley Orgánica del Servicio de Policía y del Cuerpo de Policía Nacional Bolivariana cuyo objetivo es el reordenamiento del Sistema de Policía en Venezuela (Policía, Mayo,2010). Este reordenamiento, siguiendo un nuevo modelo policial, centra sus esfuerzos en incrementar la eficiencia de la policía para garantizar mayor seguridad a la población. Esta labor se ve materializada en la creación del Cuerpo de Policía Nacional Bolivariana (CPNB).

Uno de los servicios brindados por el CPNB es el de patrullaje. Este servicio si bien se encarga de controlar y vigilar las vías de tránsito, garantizar los derechos de las personas y la seguridad en las vías públicas, no satisface plenamente las necesidades de la población. Entre los factores que afectan la buena ejecución del servicio se encuentran el desarrollo de las actividades de forma manual, la variedad de formas de recopilación de la información y la multiplicidad de normativas, criterios y procedimientos dispares. Todo esto provoca errores en el control de los procesos y la elaboración de los planes de patrullaje de manera aleatoria, lo cual degrada la calidad de los servicios brindados a la población.

La presente investigación centra su objetivo en la automatización de los procesos de la Coordinación de Vigilancia y Patrullaje (CVP), debido a que no cuenta con un sistema de gestión de información con tecnología de punta que agilice el desarrollo de las actividades llevadas a cabo. Esta idea se materializará con el diseño y desarrollo del módulo Vigilancia y Patrullaje de la personalización y ampliación del Sistema de Investigación e Información Policial (SIIPOL), que permitirá una mejor gestión de la información y abarcará los procesos que se desarrollan diariamente en la CVP, permitiendo dar una mejor respuesta a las necesidades de la población.

Abstract

The Bolivarian National Police (CPNB) is a civilian body, professional, mainly preventive. Its mission is to ensure the rights of people face situations that constitute threats, vulnerability, risk or harm to physical integrity, property, free exercise of their rights and how to control and monitor roads and transit and to facilitate conflict resolution through dialogue, mediation and conciliation in accordance with established national standards. This research focuses its overall objective for the automation of the processes of coordination that make up the organization, the Coordination of Surveillance and Patrol (CVP), because it does not have an information management system with technology that expedite the development of research carried out. This idea will materialize with the module design and development of surveillance and patrol, which will allow better management of information and cover the processes that take place daily in the CVP.



Contenido

<i>Introducción</i>	10
CAPÍTULO 1. Fundamentación Teórica	14
1.1 Introducción	14
1.2 Seguridad Ciudadana	14
1.2.1 Seguridad Ciudadana en Venezuela	14
1.3 Cuerpos Policiales	15
1.3.1 Nuevo modelo policial	15
1.3.2 CPNB	16
1.4 Vigilancia y Patrullaje	16
1.4.1 Organización de Patrullaje y Control de los procedimientos	17
1.5 Sistemas de Gestión Policial	18
1.6 Sistema de Investigación e Información Policial	21
1.6.1 Personalización y ampliación del SIIPOL	21
1.6.2 Estilos arquitectónicos	22
1.6.3 Vista arquitectónica de alto nivel	23
1.7 Ambiente de desarrollo	24
1.7.1 Metodología de Desarrollo, Lenguaje de Modelado y Herramienta CASE	24
1.7.2 Plataforma de Desarrollo, Lenguaje de Programación y Entorno de Desarrollo Integrado	25
1.8 Framework de Desarrollo	27
1.9 Sistema Gestor de Base de Datos	29
1.10 Conclusiones	30
<i>Capítulo 2. Diseño de la solución</i>	31
2.1 Introducción	31
2.2 Análisis de los medios de desplazamientos	31
2.3 Modelo de Diseño	31
2.3.1 Diagrama de Paquetes	31
2.3.2 Clases arquitectónicamente significativas	32



2.3.3 Diagrama de clases del diseño	33
2.3.4 Descripción de clases significativas	38
2.4 Modelo de Base de Datos	50
2.4.1 Diagrama de clases persistentes	50
2.4.2 Diseño de la Base de Datos.....	52
2.5 Conclusiones.....	55
CAPÍTULO 3. Implementación y Prueba de la solución propuesta.....	56
3.1 Introducción.....	56
3.2 Diagramas de componentes.....	56
3.2.1 Diagrama de componentes generales del módulo	56
3.2.2 Componentes del Dominio	57
3.2.3 Mappings	58
3.2.4 Archivos JSP.....	58
3.2.5 Subsistema Común.....	59
3.2.6 Subsistema Patrullaje	60
3.3 Pruebas de la solución.	60
3.3.2 Pruebas de Caja Negra.....	61
3.4 Conclusiones.....	63
<i>Conclusiones Generales</i>	64
<i>Recomendaciones</i>	65
Referencias Bibliográficas.....	66
Bibliografía.....	69
Anexo 1	74
CP Gestionar Planificación de Patrullaje	74



Introducción

La seguridad ciudadana es uno de los principales temas que influyen en la calidad de vida de las personas, constituyendo una preocupación constante para los organismos internacionales, los gobiernos, los mandatarios locales y del pueblo en general.

Irma Arriagada, Socióloga y Oficial de Asuntos Sociales de la Comisión Económica para América Latina y el Caribe de la ONU, define la seguridad ciudadana como la preocupación por la calidad de vida y la dignidad humana en términos de libertad, acceso al mercado y oportunidades sociales; poder disfrutar de la privacidad del hogar sin miedo a ser asaltado y poder circular tranquilamente por las calles sin temer un robo o una agresión. En este sentido América Latina se enfrenta a altos niveles de inseguridad, sin duda entre las causas más directas de los mismos se encuentra el aumento de la violencia de diversos tipos en la región y su expresión en la vida cotidiana de los ciudadanos.

La República Bolivariana de Venezuela no se encuentra ajena a estos problemas. En las últimas 3 décadas, ha transitado por varios gobiernos interesados en el enriquecimiento personal de sus líderes y de las altas esferas de la sociedad, sin tomar en serio el bienestar de las personas. Como resultado de esta situación las tasas de homicidios, por cada 100 mil habitantes, se incrementó de 11.7, a principios de la década del 80, a un 22.0 en 1995 (Arriagada, y otros, 2001).

Con la llegada de Hugo Chávez Frías a la presidencia, se ha desarrollado un proceso de transformación con el objetivo de lograr cambios políticos, económicos y sociales que garanticen la seguridad y calidad de vida del pueblo venezolano. La creación del Cuerpo de Policía Nacional Bolivariana (CPNB) es un ejemplo de las medidas tomadas en el país para lograr esos objetivos.

El CPNB es una institución adscrita al Ministerio del Poder Popular para Relaciones Interiores y Justicia (MPPRIJ) en Venezuela, comenzó a operar el 20 de diciembre de 2009 y se crea en atención a lo establecido en la Ley del Servicio de Policía y del Cuerpo de Policía Nacional Bolivariana y al deber del Estado de garantizar la seguridad ciudadana y la prevención del delito en un marco de garantía de la legalidad, del libre ejercicio de los derechos humanos, de la eficacia y la eficiencia administrativa y el



respeto a la dignidad humana (Digital). Está llamada a ser una institución profesional, estudiosa del delito, que examina el comportamiento delictivo, los factores criminógenos, utilizando tecnología de punta y aproximándose a las causas que originaron el fenómeno para promover soluciones integrales (Policía, Mayo,2010). Conformado por varias estructuras estrechamente integradas se ha trazado el objetivo de elevar la calidad del servicio que se presta a la población.

La Coordinación de Vigilancia y Patrullaje responde a estos intereses y tiene entre sus objetivos controlar las actividades relacionadas al cumplimiento del servicio de vigilancia y patrullaje, el estado de la documentación de los servicios que brindan, el cumplimiento de las actividades de preparación, el orden reglamentario en las unidades y otros aspectos generales que contribuyen al éxito de ese trabajo (Policía, Mayo,2010).

El corto período de tiempo para la implantación y consolidación de los procesos definidos en esta área, así como la diversidad en la formación de los funcionarios atentan contra los objetivos propuestos. El buen desempeño de esta área se ha visto afectado por el desarrollo de actividades de manera manual, por la variedad en formas de recopilación de la información y por la multiplicidad de funcionarios con normativas, criterios y procedimientos dispares; provocando errores en el control del proceso, y falta de uniformidad en el vocabulario técnico empleado y las actuaciones policiales ejecutadas.

Esta situación imposibilita la realización de análisis estadísticos que se pudieran generar a partir del procesamiento de la información, y por tanto arroja como resultado operaciones policiales desarrolladas de manera empírica o como reacción a las situaciones emergentes. Los efectos se evidencian, también, en la elaboración de los planes de patrullaje, los cuales se realizan de forma aleatoria ante la incapacidad de analizar información sobre criminalidad y seguridad ciudadana. Todo ello degrada la calidad de los servicios, mostrando deficiencias en el momento de dar respuesta a la población en casos de agresión o cualquier otra situación anómala.

A partir del análisis de la situación expuesta se plantea como **problema a resolver**: la organización y control de los procedimientos del servicio de vigilancia y patrullaje generan resultados no confiables para la planificación de la labor de los efectivos policiales.

El **objeto de estudio** lo constituyen los servicios que brindan los cuerpos policiales y el **campo de acción** se enmarca en el servicio de vigilancia y patrullaje en el CPNB.



El **objetivo general** de la investigación es desarrollar una aplicación informática que gestione la información de los procedimientos de vigilancia y patrullaje desarrollados por los funcionarios del CPNB.

Para darle cumplimiento se tienen los siguientes **objetivos específicos**:

- Estructurar los mecanismos que permitan establecer el intercambio de información entre la Coordinación de Vigilancia y Patrullaje y las diferentes áreas que así lo requieran.
- Integrar los componentes que se encuentran aislados y conforman el proceso definido para la elaboración de la Orden de Servicio.

La **idea a defender** es que el diseño e implementación de una aplicación informática que gestione la información de los procedimientos de vigilancia y patrullaje generará resultados confiables para la planificación del servicio.

Para dar cumplimiento a los objetivos planteados han sido definidas las siguientes **tareas de investigación**:

- Analizar los patrones de diseño y estilos arquitectónicos garantizando un diseño robusto y flexible.
- Identificar los puntos de interacción con otras áreas o estructuras garantizando el mecanismo de integración.
- Analizar la base legal existente para garantizar durante el diseño e implementación el cumplimiento de las reglas establecidas.
- Analizar la homogeneidad en los flujos identificados en diferentes cuerpos policiales para estructurar una forma genérica de ejecución.

El desarrollo de la investigación exige la utilización de métodos científicos, que permitan abordar la realidad y la naturaleza de los procesos desarrollados en la Coordinación de Vigilancia y Patrullaje (CVP), con el fin de descubrir su esencia y relaciones. Para dar respuesta a los objetivos planteados se utilizarán:

- Método Modelación, para:
 - Describir los diferentes procesos llevados a cabo en la CVP.
 - Desarrollar los modelos de diseño y de implementación.
- Método Analítico Sintético, para:



- Examinar y descomponer los flujos que toma la información dentro de los procesos definidos
- Método Histórico-Lógico, para
 - Analizar la documentación relacionada a los Sistemas de Gestión de Información Policial viendo su evolución y funcionamiento.

El empleo de diferentes técnicas investigativas permite mejorar las labores de obtención de datos que enriquezcan la investigación. Entre las técnicas utilizadas está la entrevista, para mediante reuniones, con diferentes especialistas, obtener información relevante para el desarrollo de la investigación.

La investigación quedará estructurada en 3 capítulos que agruparán los contenidos de la siguiente manera:

Capítulo 1. Fundamentación Teórica: se analizan elementos teóricos de la investigación tales como Gestión de Información Policial, procesos de vigilancia y patrullaje, metodología, lenguaje y herramientas de desarrollo empleadas.

Capítulo 2. Diseño de la solución propuesta: se presenta el modelo de diseño y el modelo de base de datos desarrollado.

Capítulo 3. Implementación y Prueba de la solución propuesta: se presentan los diagramas de componentes así como el diseño y ejecución de las pruebas desarrolladas.



CAPÍTULO 1. Fundamentación Teórica

1.1 Introducción

A lo largo de la investigación serán abordados temas relacionados con la gestión de la información policial y la seguridad ciudadana. Se analizarán los principales conceptos relacionados con el módulo a implementar, para lograr una mayor comprensión de los temas relacionados con el problema a resolver. Se hará un análisis de algunos sistemas similares al que se ha de desarrollar.

Se realizará, además, un estudio del ambiente de desarrollo, brindando las principales características de la metodología, lenguaje y herramientas que se utilizarán en la implementación de la solución.

1.2 Seguridad Ciudadana

La seguridad ciudadana toma el individuo como elemento central de su definición; y esgrime sus argumentos considerando las garantías generales inherentes al mismo, ya sean percibidas de forma directa o indirecta. La misma implica que los ciudadanos estén en situación de vivir y convivir disponiendo de una protección necesaria tal que les permita superar los peligros propios de un entorno social riesgoso. Aun cuando en la práctica dicho entorno va a proporcionar siempre distintos grados de inseguridad derivados de la acción de personas, grupos e instituciones o de elementos del medio natural que amenacen la vida, la integridad física o de los bienes de las personas (Soto). La concreción o el logro de este tipo de seguridad, se entiende que es de responsabilidad tanto de la policía como de los ciudadanos mismos.

1.2.1 Seguridad Ciudadana en Venezuela

La falta de seguridad es un problema latente en Venezuela, de acuerdo con las cifras reportadas por El Cuerpo de Investigaciones Científicas Penales y Criminalísticas. División de estadística –CICPC-, en el año 2005 en el país ocurrieron 10.098 homicidios, lo que determina una tasa de 38 homicidios por cada 100.000 habitantes, que lo sitúa con una violencia superior a Colombia, Jamaica, Brasil y México y solamente superada por Honduras y el Salvador (Velásquez, 2006).



El Estado venezolano asume el compromiso de enfrentar esta situación de violencia partiendo de una política sostenida, proyectada en el tiempo y espacio; concebida en términos científicos no sólo como política anti delictiva, sino con el alcance de la moderna política criminal y el consenso del control social formal e institucionalizado.

La Constitución de la República Bolivariana de Venezuela, aprobada en 1999, mediante el artículo 322 establece la organización de los órganos de seguridad ciudadana, como medio para garantizar la protección de los ciudadanos y sus hogares en el disfrute de los derechos fundamentales, incorpora la creación de instituciones e instrumentos legales que permitan abordar integral y eficazmente la problemática de la inseguridad ciudadana. Sin embargo este es un fenómeno social, que se ha venido incrementando en los últimos años a pesar de los múltiples esfuerzos realizados por el Estado para disminuir sus consecuencias.

1.3 Cuerpos Policiales

Un cuerpo policial es una organización estatal que se encarga de mantener el orden público y garantizar la seguridad de los ciudadanos de acuerdo a las órdenes de las autoridades políticas (Definición). Para lograr esto deben velar por el cumplimiento de la Ley, garantizar la protección de las personas y sus bienes. Mantener el orden público y luchar contra la delincuencia, la investigación de delitos, así como el patrullaje sistemático y preventivo, son otras de las funciones de estas instituciones (Pública).

Un estudio realizado por la Comisión Nacional para la Reforma Policial (CONAREPOL), en Venezuela, muestra dos características importantes en los 138 cuerpos policiales existentes en ese país en el año 2010. Por un lado, su poca regulación, pues la mayoría de los cuerpos no tienen manuales de procedimientos o de ingreso formalizados, cuando existen son desconocidos por los propios funcionarios y casi nunca se obliga a su aplicación. Por otro lado, se trata de unas instituciones militarizadas, con unos rasgos fuertemente castrenses: la mayoría de policías están dirigidas por militares activos o retirados y la estructura organizativa dividida en oficialidad y tropa, los mecanismos de formación, la práctica de acuartelamiento, la forma de disciplina y la simbología son todas de tipo militar y no se compadecen con las prácticas de una policía ciudadana (Emanuelsson, 2010).

1.3.1 Nuevo modelo policial



Ante la situación existente en los cuerpos policiales se impuso realizar una reforma policial que tuvo como propósito dos metas centrales: por un lado, la necesidad de incrementar la eficiencia de la policía para garantizar mayor seguridad a la población y, por otro, poner correctivos a sus excesos e ilegalidades. Fue necesario reformar la policía para convertirla en una institución más eficiente en su tarea de hacer cumplir la ley, pero también para obligarla a ella misma a ser más cumplidora de la norma. Se hizo un diagnóstico nacional de cada uno de los cuerpos. Se obtuvo la información de lo que estaba bien lo que estaba mal, y se hizo un proceso de estudio al detalle de los problemas más graves, acompañado también de una consulta popular. Entre el diagnóstico y la consulta salió el Nuevo Modelo de Policía (Emanuelsson, 2010). Este nuevo modelo apunta a las distintas dimensiones, no solo a la formación, al régimen disciplinario de previsión social o al tema de normas de actuación, o al uso de la fuerza, sino que el nuevo modelo abarcaba todos los ámbitos de la policía. Se propuso y aprobó una nueva ley que organizara todo el servicio de policía y estableciera cual era el nuevo modelo policial.

1.3.2 CPNB

La propuesta del nuevo modelo se concreta en la concepción, creación y despliegue del CPNB, el cual no sustituye las policías estatales o municipales, sino que las complementa de acuerdo a criterios y niveles de actuación establecidos en ley. De esta manera se conforma un cuerpo policial que trabaja bajo principios, formación básica de los funcionarios y estándares de procedimientos unificados. El CPNB es una institución profesional, estudiosa del delito, que examina el comportamiento delictivo, los factores criminógenos, utilizando tecnología de punta y aproximándose a las causas que originaron el fenómeno para promover soluciones integrales (Policía, Mayo, 2010).

Son varias las estructuras que componen el CPNB y disímiles los elementos que se conjugan en el logro de los objetivos; pero sin lugar a dudas uno de los factores fundamentales para garantizar la reducción de los índices de delincuencia, lo constituye la calidad del servicio brindado a la población por parte de la CVP.

1.4 Vigilancia y Patrullaje

Uno de los principales servicios brindados por esta coordinación es el de vigilancia y patrullaje sistemático. La definición de estos términos se enmarcan en la labor policial analizada; refiriéndose a la acción que



realizan los funcionarios de los cuerpos de policía al recorrer un territorio previamente determinado y teniendo como objetivos, ofrecer seguridad a las entidades e instituciones, garantizar el buen orden de convivencia y obstaculizar la ocurrencia de hechos delictivos (Policía, 2010).

Dado que la capacidad preventiva está determinada por la presencia física de un efectivo antes de la ocurrencia de un hecho determinado, resulta evidente la obligación de organizar a los funcionarios entorno a un servicio acorde a las necesidades reales. Este tiene entre sus principales objetivos mantener un adecuado clima de seguridad y protección a las personas y sus bienes; desarrollar y consolidar relaciones armónicas y estrechas con la ciudadanía, con el fin de estimular su participación consciente en la prevención de la actividad delictiva.

El patrullaje puede ser clasificado, según el medio de desplazamiento utilizado, en patrullaje vehicular, motorizado o a pie (Policía, Mayo, 2010).

- **Patrullaje vehicular:** proyectado para la cobertura de amplias áreas mediante recorridos en vehículos especialmente diseñados para tal fin y brindar una priorizada atención a los requerimientos de la ciudadanía en el sector asignado.
- **Patrullaje en moto:** proyectado para realizar un rápido desplazamiento y concurrencia hacia el lugar de los hechos, complementar y cooperar con el resto de los servicios y subsistemas. Su ventaja principal es que con la moto el oficial se puede trasladar a lugares que en una unidad vehicular no podría y a lugares que a pie o en vehículo tardaría mucho en llegar (lugares accidentados o estrechos, congestionamientos de vehículos, etc.).
- **Patrullaje a pie:** proyectado para que los funcionarios en la vía pública cumplan con la misión de realizar la vigilancia y el patrullaje o la regulación vial en un área predeterminada sin el empleo de ningún medio de transporte. Su ventaja principal radica en que se establece un mayor contacto entre el oficial y la comunidad y, puede ver y escuchar de cerca estímulos y sucesos que en vehículo o motor no podría, además este contacto origina una relación que da acceso a información importante para la policía.

1.4.1 Organización de Patrullaje y Control de los procedimientos

Para garantizar un mejor servicio es necesario lograr una buena organización de los mismos y garantizar el control de todos los procedimientos ejecutados durante un turno de patrullaje. Lo primero se logra



mediante una adecuada gestión y control de las Órdenes de Servicios y lo segundo llevando a cabo un control de las Actuaciones Policiales ejecutadas por los funcionarios.

Una Orden de Servicio (OS) no es más que la distribución de los funcionarios de un Grupo de Patrullaje (GP) por áreas y sectores en un Turno de Patrullaje (TP) determinado. Para lograr esto se comienza con la creación de los GP, que consiste en la agrupación de patrulleros según un tipo de servicio determinado (Patrullaje a Pie, Vehicular o Motorizado). Luego a cada grupo se le realizan Planificaciones de Patrullaje (PP), proceso que consiste en distribución previa de los GP en los turnos establecidos para la ejecución del servicio. Durante el proceso de creación de la OS, los funcionarios son distribuidos en Unidades de Patrullaje (UP), las cuales constituyen el elemento básico operativo de despliegue policial. Su configuración varía según el medio utilizado para su desplazamiento. Una unidad de patrullaje está conformada por al menos dos funcionarios policiales y el medio de transporte asignado. Esta OS contiene toda la información relacionada con el servicio a prestar durante el TP. Luego de realizado el servicio se registra toda la información relacionada con las Actuaciones Policiales (AP) ejecutadas durante el mismo. Estas AP son las actuaciones cotidianas, eventuales, sistemáticas y coordinadas realizadas por las funcionarias y los funcionarios policiales en el ejercicio de sus funciones, en asuntos que así lo justifiquen (Policía, 2010).

1.5 Sistemas de Gestión Policial

Para lograr la gestión y control de estos procesos desarrollados por la CVP es necesario contar con un sistema informático que les permita realizarlos de manera eficiente y segura.

Un Sistema de Gestión de Información (SGI) no es más que un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades que se realizan en una organización o empresa. Se utilizan para automatizar procesos y dar apoyo a otras acciones, con el objetivo de mejorar el tiempo de respuesta a las necesidades de los clientes u otras entidades, para acceder concurrentemente a documentos e información, minimizar el tiempo de acceso a información crucial, así como al análisis de estadísticas y resultados de investigaciones para la toma de decisiones, generar documentos automáticamente y dar solución a la problemática planteada, reducir riesgos de pérdida de documentos de vital importancia para la empresa, proporcionando grandes beneficios en todas las áreas de la misma (Cohen, y otros, 2000).



Un Sistema de Gestión Policial (SGP) es, entonces, un SGI que se especializa en automatizar los procesos en entidades policiales, cuya característica representativa es la confidencialidad, empleando un sistema de trabajo que facilita y soporta la generación de estrategias de prevención y combate contra el crimen, mediante el análisis sistemático de la información y la instrumentación de mecanismos efectivos de evaluación de resultados. Una correcta preparación del personal de las instituciones policiales y el uso de un sistema eficiente de gestión policial es la combinación perfecta para combatir los males que ponen en peligro la seguridad social y el orden interior de cada país.

En la actualidad a nivel mundial está en aumento la tendencia al uso de estos sistemas, la cual está dada por la necesidad de centralizar la información e integrar el trabajo conjunto con el fin de reducir innecesarios y molestos trámites generados durante el esclarecimiento de los hechos, disminuyendo considerablemente errores en los procesos investigativos, así como aumentando la capacidad y disponibilidad de la información. Algunos de estos sistemas son:

- **Sistema Territorial de Emergencias y Gestión Policial (STEGPOL)**, perteneciente a la República de Chile, es un SGP cuya idea es integrar a las diferentes entidades como Carabineros, Investigaciones, Ministerio del Interior y Municipios, entre otros, en una Plataforma Nacional Común de Información que permita el intercambio de datos y que sirva de apoyo a la gestión operacional regional o comunal, donde dichas Instituciones estén interconectadas entre sí. (Mapas Digitales, S.A)
- **Sistema de Gestión Policial (SIGEPOL)**, que fue definido como un sistema que tiene como objetivo la captura de la información sobre la gestión policial, que sirva de base a la toma de decisiones en la sala situacional del Centro de Tratamiento y Análisis de Información de Seguridad Ciudadana (CTAISC), a fin de que la información y el análisis de esta ayuden a combatir el delito en la República Bolivariana de Venezuela.
- **Sistema de Investigación e Información Policial (SIIPOL)** es un sistema que es utilizado por el Cuerpo de Investigaciones Científicas Penales y Criminalísticas (CICPC), de la República Bolivariana de Venezuela, para realizar actividades de investigación policial, que ayuda a controlar y organizar el trabajo en las dependencias del CICPC y mejora el nivel de respuesta a las necesidades de seguridad del ciudadano venezolano. Los objetivos fundamentales del SIIPOL son tener centralizada toda la información necesaria de modo que el proceso de investigación sea lo



más ágil posible. Aumentar el nivel de comunicación entre las diferentes áreas. Disminuir el trabajo manual que se realiza para resolver casos. Proporcionar información actualizada de las personas, vehículos, objetos y armas relacionadas con algún delito. Brindar información a entidades del gobierno y la población en general.

- **Sistema Automatizado del Puesto de mando de Seguridad Pública** perteneciente a la Unidad Provincial de Patrulla (UPP) de Ciudad de la Habana. Este es un centro de gestión a emergencias que cuenta con un sistema informático el cual se explota desde hace más de diez años con el objetivo de garantizar la tranquilidad ciudadana en la Capital de Cuba. Dentro de los procesos fundamentales con que cuenta el sistema informático se pueden destacar los referentes a las Llamadas de Emergencia, conocido también como el servicio 106, el proceso de la Dirección Operativa de las patrullas que se mantienen en la vía y la supervisión de estos procesos. El organismo cuenta además con un Sistema Integral de Trabajo de la Unidad de Patrullas (SITUP), de donde se obtienen datos de interés a la hora de tomar decisiones. Cuenta además con áreas de atención y áreas de patrullaje donde se mantiene un constante seguimiento de las diferentes acciones que se realizan mediante el mapa, lo que posibilita un estricto control y supervisión del trabajo de los diferentes recursos que se encargan de garantizar el bienestar de la población. Cuenta también con varios procesos de apoyo entre los que se encuentran la Creación de Áreas, Planes, Servicios y Variante Operativa, los cuales están formados por una serie de tareas necesarias para la ejecución de los procesos claves de la organización. En este sistema se registran procesos con cierta similitud a los que se desean implementar para el CPNB pero no se adaptan a las necesidades de la CVP aunque el proceso de la creación de áreas para dividir el territorio para la ejecución del patrullaje debe ser tomado y adaptado para el CPNB. En el caso de la CVP lo que se realizará será la división del territorio en áreas, sectores y comunidades. El adoptar esta característica para el sistema permitirá distribuir equitativa y eficazmente los recursos policiales para garantizar la presencia de unidades policiales cerca de los sitios de mayores ocurrencias delictivas.

Luego de analizar estos sistemas se puede que todos resuelven los problemas para los que fueron originalmente creados, pero ninguno se adapta a las necesidades del CPNB.



1.6 Sistema de Investigación e Información Policial

El CPNB necesita contar con un SGP que le facilite el trabajo con la información que se genera en todos los procesos que esta realiza. Por otro lado el CICPC es una institución similar al CPNB que, como ya se ha mencionado, cuenta con su propio sistema, el SIIPOL.

Estas son dos entidades policiales con procesos homogéneos en su funcionamiento pero totalmente heterogéneos en cuanto a los procedimientos empleados. La realización de la labor articulada entre el CICPC y CPNB, con el empleo de métodos modernos de investigación y análisis, permitirá consolidar la lucha por el disfrute de los derechos ciudadanos. Esta premisa presupone una integración en el trabajo realizado con el objetivo de garantizar un desempeño policial de excelencia.

La Policía Nacional Bolivariana surge con la capacidad de desarrollar todos los procesos policiales convencionales y cuanto procedimiento adicional tribute a un desempeño policial de excelencia. Entre estos se encuentran: Detención y reseña de detenido, Recepción de denuncia, Planificación de operativo policial, Gestión de funcionario y armamento, Investigación penal, Investigación criminalística, Investigación forense. Policía comunal, Atención a la víctima y Vigilancia y patrullaje.

1.6.1 Personalización y ampliación del SIIPOL.

El actual SIIPOL no se adapta a la configuración estructural y, además, no satisface todas las necesidades del CPNB. Sin embargo tiene ya implementado muchos de las funcionalidades que son requeridas por el mismo. Por lo tanto es preciso personalizar este sistema para adaptarlo a las necesidades del CPNB y, además, desarrollar las nuevas funcionalidades. Los nuevos módulos a desarrollar son los de Policía Comunal, Atención a la Víctima, Actuación Policial y el de Vigilancia y Patrullaje, siendo en este último en el que se centra el presente trabajo. Sin embargo para el desarrollo de éste módulo se hace necesario adaptar el sistema a las necesidades del mismo. Una de estas necesidades es que el sistema debe ser capaz de comunicarse con el Sistema de Gestión de Mantenimiento Vehicular (SGMV), este sistema es el que controla el estado de los medios de transporte de los que disponen las dependencias policiales y que deben ser usados para la ejecución del servicio de patrullaje. Por otro lado para la ejecución del patrullaje cada funcionario debe tener un radio portátil y en el



actual SIIPOL no se gestionan los mismos, por lo cual es necesario incluir dentro de los CU de Dotación de Equipos Policiales la gestión y consulta de los radios portátiles asignados a una dependencia policial.

1.6.2 Estilos arquitectónicos

El SIIPOL al ser un sistema que ya lleva varios años de desarrollo y que se encuentra en explotación por parte del CICPC tiene ya definidos sus estilos y patrones arquitectónicos, los cuales se adaptan a sus necesidades.

Un estilo arquitectónico define una categoría del sistema que abarca un conjunto de componentes que realizan una función requerida por el sistema, un conjunto de conectores que posibilitan la coordinación, comunicación y entre los componentes, así como restricciones que definen como se integran los componentes para formar el sistema (Pressman).

1.6.2.1 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

1.6.2.2 Arquitectura en Capas

El SIIPOL usa una arquitectura en capas permitiendo desarrollar un sistema modular, escalable y de fácil desarrollo y mantenimiento, debido a las ventajas que está brinda, entre las que se encuentran una reutilización real del código, facilita la estandarización, las dependencias se limitan a las capas adyacentes lo que garantiza una contención de cambios a una o pocas capas (Pressman).

1.6.2.3 Patrón Modelo Vista Controlador (MVC)

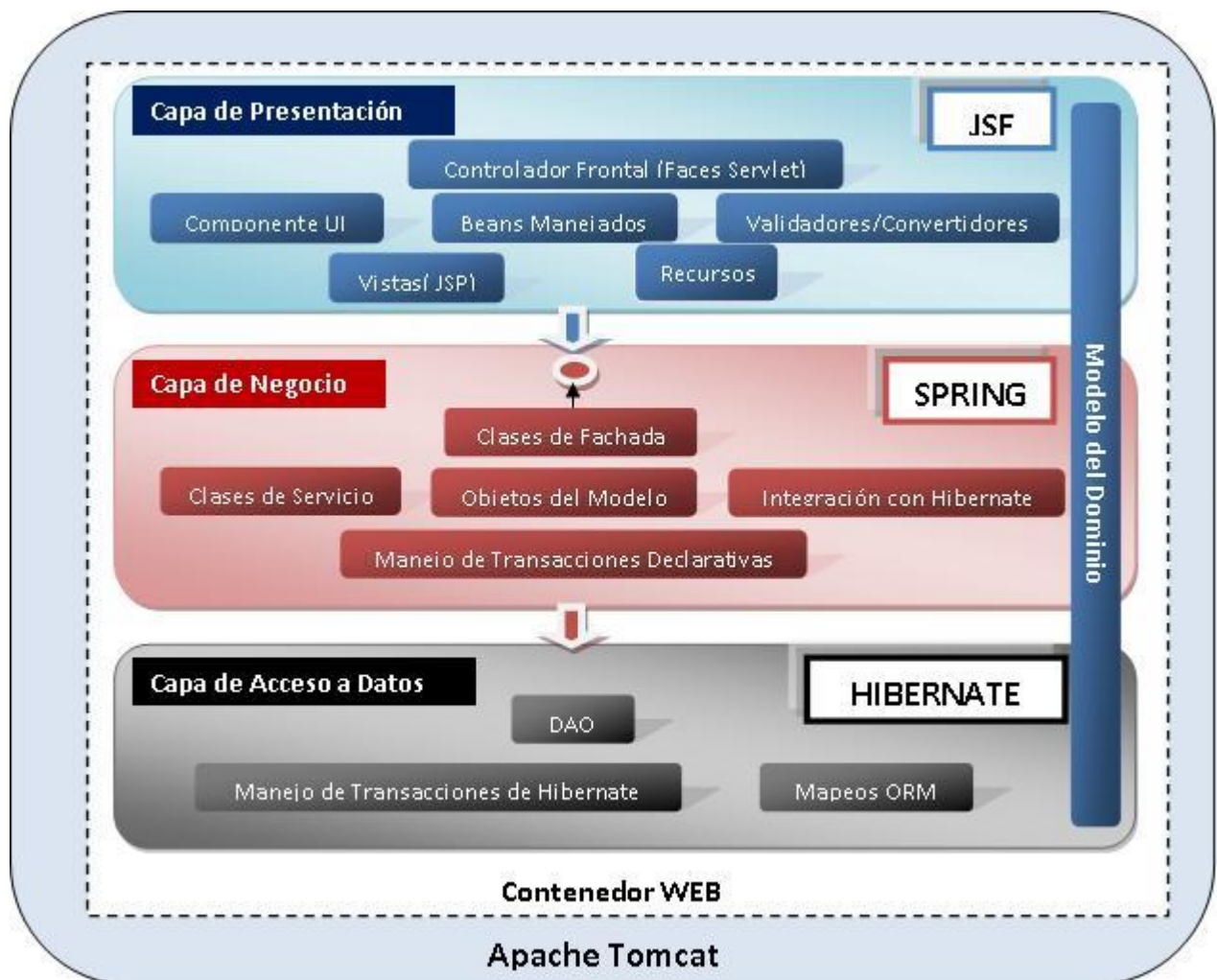
El patrón arquitectónico utilizado es MVC (Modelo Vista Controlador), el que permite una serie de ventajas a partir de su propia estructura: el modelo, que es la representación de la información que maneja la aplicación; la vista, que representa el modelo de forma gráfica disponible para la interacción con el usuario y el controlador que es el encargado de manejar y responder las solicitudes del usuario, procesando la



información necesaria y modificando el modelo en caso de ser necesario. Entre las utilidades que brinda este patrón se encuentran la separación del modelo de la vista, es decir, separar los datos de la representación visual de los mismos, haciendo mucho más sencillo agregar múltiples representaciones de los mismos datos o información (Pressman).

1.6.3 Vista arquitectónica de alto nivel

Las funcionalidades a implementar como parte del Módulo Vigilancia y Patrullaje se acoplan a la arquitectura definida por el SIIPOL. Esta arquitectura se muestra en la siguiente figura.





1.7 Ambiente de desarrollo

Debido a que el módulo a desarrollar formará parte de la personalización y ampliación del SIIPOL las metodologías, herramientas y lenguajes a utilizar durante el desarrollo de la solución son las usadas por el equipo de desarrollo original del SIIPOL. Las mismas se adaptan perfectamente a las necesidades del módulo de Vigilancia y Patrullaje.

1.7.1 Metodología de Desarrollo, Lenguaje de Modelado y Herramienta CASE

Durante el desarrollo de software, muchas tareas y actividades se tornan engorrosas, trayendo esto consigo que el proceso de desarrollo de software se convierta en riesgoso y difícil de controlar, obteniendo con esto clientes insatisfechos con el resultado final. La forma para solucionar o tratar de llevar a cabo un eficiente desarrollo es aplicando una metodología de desarrollo de software, la cual es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software, indicando quién debe hacer cada actividad, cuándo hacerla y qué debe hacer (Jacobson, y otros, 2007).

El Proceso Unificado de Desarrollo (RUP) es uno de los procesos más generales de los existentes actualmente, ya que está pensado para adaptarse a cualquier proyecto y para conseguir sus objetivos se basa fundamentalmente en el orden y la documentación. RUP se caracteriza por ser dirigido por casos de uso, centrado en la arquitectura y ser iterativo e incremental, donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo (Jacobson, y otros, 2007).

Es importante señalar que RUP ha sido seleccionado teniendo en cuenta la magnitud de la propuesta de solución que hace que las partes a entregar deban quedar bien documentadas internamente, la cantidad de miembros del equipo, el tiempo con que se dispone para su culminación así como lo distante que se encuentran desarrolladores y clientes geográficamente. Por otra parte la experiencia de los desarrolladores es muy poca para enfrentar un proyecto de esta magnitud, sin pautas bien definidas y una referencia disponible en todo momento.

En el proceso de desarrollo de software el modelado ejerce un papel fundamental. RUP utiliza como lenguaje de modelado el Lenguaje Unificado de Modelado (UML) que con la utilización de diagramas y gráficos brindan una mejor perspectiva de lo que se quiere. El mismo se utiliza para definir un sistema de



software, detallar los artefactos que se generan durante el desarrollo de software, documentar y construir; de esta forma, ofrece una amplia vista del sistema, desde la cual se logra un entendimiento entre todos los vinculados al desarrollo de software (Jacobson, y otros, 2007).

UML ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales, tales como procesos de negocios y funciones del sistema, y aspectos concretos, como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Captura decisiones y conocimiento sobre los sistemas que se deben construir. No define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Da apoyo a la mayoría de los procesos de desarrollo orientados a objetos. En fin UML es el enlace entre quién tiene la idea y el desarrollador, la comunicación es su principal objetivo (Larman, 1999).

Las herramientas para la Ingeniería de Software Asistida por Computadoras (CASE por sus siglas en inglés) de modelado con UML permiten representar el software mediante diagramas que se generan durante las diferentes etapas del proyecto. Permiten además hacer análisis y diseño orientado a objetos y facilitan el desarrollo del proceso. La herramienta definida fue Visual Paradigm for UML 6.4 Enterprise Edition, debido a que es una herramienta que soporta el ciclo de vida del desarrollo de software: el Análisis y Diseño, Construcción, Pruebas y Despliegue (Organización, 2009).

Visual Paradigm es un producto de calidad, que soporta aplicaciones Web. Su diseño está centrado en casos de uso y enfocado al negocio generando un software de mayor calidad, presenta capacidades de ingeniería directa e inversa y disponibilidad en múltiples plataformas. Permite el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación. Es muy fácil de usar en la creación de todo tipo de diagramas UML, para los que dispone de un número considerable de estereotipos que permiten un mayor entendimiento de los mismos. Visual Paradigm para UML soporta un conjunto de lenguajes, tanto para la generación de código como para la realización de ingeniería inversa en: Java, C++, CORBA IDL, PHP, XML Schema, Ada y Python. La ingeniería inversa también soporta clases Java, .NET dll y .exe, JDBC, así también los archivos de mapeo Hibernate. Visual Paradigm facilita generar código Java a partir de modelos y también generar modelos a partir de código Java, y de esta forma cualquier cambio que se realice en el código se refleja en el modelo y viceversa (Organización, 2009).

1.7.2 Plataforma de Desarrollo, Lenguaje de Programación y Entorno de Desarrollo Integrado



En el desarrollo de software cuando se desea seleccionar un lenguaje de programación se debe tener en cuenta elementos claves los cuales son: el entorno de ejecución, el rendimiento, la escalabilidad, la portabilidad y la seguridad. Teniendo en cuenta estos puntos se puede seleccionar el lenguaje que más se ajuste según lo que se desea.

La plataforma Java 2 Enterprise Edition (J2EE) creada por Sun Microsystems define un conjunto de estándares y especificaciones para el desarrollo de aplicaciones empresariales basadas en la tecnología Java. (Objetos, 2002) La plataforma como tal es una máquina virtual encargada de la ejecución de aplicaciones, y un conjunto de librerías estándar que ofrecen funcionalidad común. J2EE basa su funcionamiento en un procesador virtual que ejecuta programas escritos en un lenguaje de programación; dicho procesador es la máquina virtual de Java o Java Virtual Machine (JVM) que traduce el bytecode en instrucciones nativas de la plataforma destino (Objetos, 2002). J2EE no es solo una plataforma o una tecnología, sino un estándar de desarrollo, construcción y despliegue de aplicaciones. Provee soporte para múltiples sistemas operativos y tiene una independencia total de la arquitectura de hardware.

J2EE está capacitada para ejecutar aplicaciones creadas usando el lenguaje de programación Java. Java como lenguaje de programación es multipropósito, reúne todas las características de un ambiente orientado a objetos: es sencillo, cuenta con capacidad de generación de aplicaciones distribuidas, robusta, segura, de arquitectura neutral, portable, multihilo, dinámico y de alto rendimiento. En general, cualquier aplicación que se desee hacer con acceso a través de la web se puede desarrollar utilizando Java (Alvarez, 2001). Permitirá reutilizar códigos, lo que beneficiará al desarrollador reduciendo el trabajo mecánico y posibilitando una codificación rápida y fácil de entender.

El lenguaje de programación por sí solo no es suficiente para el desarrollo de la aplicación, por lo que es necesario contar con un Entorno de Desarrollo Integrado (IDE). Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDE pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes (Oviedo).

Eclipse Helios Service Release 1 es un IDE que cuenta con herramientas que ofrecen un modelo de programación unificado, además de muchas otras que permiten desarrollar y depurar aplicaciones J2EE, simplifica la ejecución y depuración de dichas aplicaciones. Proporciona un entorno de desarrollo Ajax basado en RichFaces y un excelente soporte para poderosos frameworks como JSF, Spring, Hibernate, entre otros (LaFlecha, 2007). Eclipse Helios permite una codificación cómoda a los desarrolladores, a



través de las diferentes perspectivas en las que organiza los elementos en el marco de trabajo, el cual está diseñado para maximizar la productividad de los desarrolladores. Posee una poderosa ayuda en línea para el trabajo con las diferentes librerías de clases que le aportan a los desarrolladores valiosas informaciones ante una situación dada. La integración con el IDE de algunas herramientas como el subclipse 1.6.2, permitirá sincronizar el proyecto con un repositorio; subir, actualizar o revertir cambios; ignorar contenido y manejar diversas ramas del proyecto.

1.8 Framework de Desarrollo

El uso de framework para el desarrollo de la aplicación ayuda a construir una arquitectura sólida, robusta, brindando consistencia al código y facilitando la integración de nuevas funcionalidades. Básicamente un framework (término en inglés para marco de trabajo) es una estructura de soporte mediante la cual puede ser desarrollado un proyecto de software. Generalmente incluye programas, bibliotecas y un lenguaje interpretado que optimizan y aceleran el proceso de desarrollo del software. Son desarrollados para brindar a los programadores y diseñadores una mejor organización y estructura a sus proyectos. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo tales como el uso de patrones.

Capa de presentación

El framework Java Server Faces (JSF) es un framework para construir interfaces de usuario en aplicaciones web del lado del servidor, basadas en tecnología Java, pero además se basa en el patrón Modelo Vista Controlador (MVC), utiliza Java Server Pages (JSP) como motor de plantillas. JSF está compuesto por: componentes UI (User Interface Components) definidos por etiquetas y XML, eventos y validadores. Posibilita la reutilización de código, la separación de roles y la facilidad de uso de las herramientas, además tiene como meta hacer el desarrollo web más rápido y fácil. Permite almacenar automáticamente la información de los formularios, actualizando el mismo en el momento de ser mostrado al cliente, encapsula la lógica de manipulación de los eventos y la forma en que se muestran los componentes ya definidos. (Torrijos) Permite que los desarrolladores piensen en términos de componentes, eventos, backing beans y otras interacciones, en vez de peticiones, respuestas y marcas. (Ramón, 2006)

JSF permite la conexión de eventos generados en el cliente a código de la aplicación en el lado del servidor, mapear componentes UI a una página de datos del lado del servidor, construir una UI con



componentes reutilizables y extensibles, guardar y restaurar el estado de la UI más allá de la vida de las peticiones de servidor. De forma general facilita el trabajo de construcción y mantenimiento de aplicaciones Web con UIs del lado del servidor. (Torrijos, 2003)

Además, existen librerías específicas de gran importancia para JSF, una de ellas es la RichFaces que posee una amplia librería de componentes visuales y Ajax4jsf que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario, etc. En definitiva Ajax4jsf permite dotar a nuestra aplicación JSF de contenido mucho más profesional con muy poco esfuerzo.

Capa de lógica del negocio

Spring Framework (también conocido como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. Ofrece un amplio soporte a frameworks como Java Server Faces (JSF), se integra con frameworks de acceso a datos como Hibernate, ofreciendo un manejo seguro y eficiente de sus sesiones y manejando la configuración de la SessionFactory y las fuentes de datos JDBC en el contexto de la aplicación. (Piñeiro, y otros, 2006) Su configuración está basada en XML.

Los principales módulos de este framework son:

- **Core:** Como su nombre indica, es el núcleo de Spring. Permite técnicas de Inversión del Control (IoC) como la inyección de dependencias. De esta forma permite inyectar las dependencias de un bean en el momento de su creación utilizando un manejador externo, mediante esta técnica la IoC promueve el bajo acoplamiento de las clases. (Walls, y otros, 2005)
- **Context:** Proporciona herramientas para acceder a los beans y da soporte a propagación de eventos, resourcebundles, carga de recursos y creación transparente de contextos por parte de los contenedores.
- **DAO:** Proporciona una capa de abstracción JDBC (Java Database Connectivity) y una forma de administrar transacciones.
- **ORM:** Provee capas de integración para APIs de mapeo objeto-relacional.
- **AOP:** Proporciona una implementación de programación orientada a aspectos, permitiendo definir puntos de corte e interceptores.



- Web: Provee de características de integración orientadas a la web, como funcionalidad multiparte, inicialización de contextos mediante servlet listeners y un contexto de aplicación orientada a la web. También permite integrar de forma sencilla otros frameworks como Struts, JSF o WebWork.
- Spring MVC: Provee una implementación Modelo-Vista-Controlador que permite el uso del resto de las funcionalidades del Spring Framework.

Spring viene integrado con un framework de seguridad, Acegi Security, que gestiona todo el mecanismo de autenticación y autorización. Provee un sistema de validación que no está ligado a una capa concreta, si no que puede ser usado en cualquier capa de la aplicación, se encarga del manejo de errores.

Capa de acceso a datos

La persistencia de datos de cualquier aplicación de software ha sido uno de los aspectos más importantes que influye directamente en el desempeño y tiempos de desarrollo de las mismas.

Hibernate es una poderosa herramienta de mapeo objeto-relacional Open Source, que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación mediante archivos XML. Permite desarrollar clases persistentes siguiendo las pautas de la programación orientada a objeto tales como asociaciones, herencia, polimorfismo, composición y manejo de colecciones. Posee un lenguaje propio de consultas HQL (Hibernate Query Language) e incorpora una API mediante la cual se pueden efectuar consultas programáticamente llamadas "criteria"; todo esto sin quitar el uso del SQL nativo. Brinda soporte para el manejo de transacciones, concurrencia y caché. Ofrece soporte para un amplio conjunto de colecciones de Java. Es no intrusivo. Puede integrarse en la arquitectura J2EE y permite configurarlo con JDBC. Libera al desarrollador de las preocupaciones del tipo de motor de base de datos, así como el lenguaje nativo a usar en cada caso, aparte de que facilita trabajar limpiamente en un modelo objetual, aporta bajo acoplamiento entre negocio y persistencia. Hibernate simplifica el proceso de creación de aplicaciones y reduce drásticamente el tiempo de desarrollo, mantenimiento y resolución de errores.

1.9 Sistema Gestor de Base de Datos

Oracle es un sistema gestor de base datos que cuenta con un gran prestigio a nivel mundial. Es considerado como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.



Es una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. Posee un lenguaje de diseño de bases de datos PL/SQL, un lenguaje de 5ta Generación, muy potente para tratar y gestionar la base de datos, y muy versátil, que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa muy fuerte. Permite el uso de particiones para la mejora de la eficiencia, de recopilación e incluso ciertas versiones admiten la administración de bases de datos distribuidas. El software del servidor puede ejecutarse en multitud de sistemas operativos.

Oracle es ventajoso para el desarrollo del sistema ya que ha sido diseñado para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.

1.10 Conclusiones

Una vez expuestos los detalles relacionados con la fundamentación teórica de la investigación se puede concluir que:

- Ninguno de los SGP analizados satisface las necesidades de la CVP, por lo tanto se hace necesario implementar uno que de solución al problema planteado.
- Las características del entorno son favorables para que el SIIPOL se adapte a las necesidades del CPNB, por lo tanto el sistema a implementar se integrará como un nuevo módulo de la personalización y ampliación del SIIPOL.
- El análisis de las tecnologías, arroja un paquete de herramientas, framework y librerías, que debidamente empleadas y combinadas ofrecen la mejor solución para obtener un producto de calidad.



Capítulo 2. Diseño de la solución

2.1 Introducción

Una vez que han sido definidas las metodologías y herramientas a utilizar en el desarrollo del software se puede dar paso al diseño de la solución.

2.2 Análisis de los medios de desplazamientos

Un elemento particular del módulo a desarrollar es que para la creación de la Orden de Servicio es necesario obtener un listado de los medios de transporte que no estén en mantenimiento. Esto implica la necesidad de que el sistema se comunice con el Sistema de Gestión de Mantenimiento Vehicular (SGMV) para obtener de cada Dependencia los medios que estén disponibles. No es objetivo de este trabajo la forma de interacción entre las aplicaciones, pues el equipo de arquitectura se encarga de la comunicación entre las mismas. Sin embargo es necesario tener una independencia total entre la lógica de negocio del módulo y los mecanismos de obtener los datos de los medios de transporte. Para ello se creará una clase separada y se le asignará la responsabilidad de obtener los datos de mantenimiento vehicular, para de esta forma aislarla del resto de las clases y así disminuir el impacto de cualquier cambio en esta área.

2.3 Modelo de Diseño

2.3.1 Diagrama de Paquetes

Los paquetes son utilizados para estructurar el modelo de diseño mediante su división en partes más pequeñas y agrupar elementos relacionados de dicho modelo con propósitos organizacionales. El criterio de empaquetamiento escogido fue agrupar los casos de uso que dan soporte a un determinado proceso de negocio; con el objetivo de lograr la alta cohesión y el bajo acoplamiento.

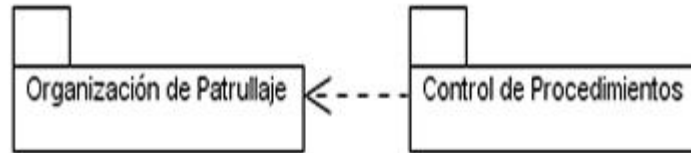


Fig. 1 Diagrama de Paquetes del módulo Vigilancia y Patrullaje.

Paquete Organización de Patrullaje:

En este paquete se encuentran los casos de uso (CU) relacionados con la gestión de la organización del servicio de vigilancia y patrullaje: Grupos de Patrullaje, la Planificación y las Órdenes de Servicio asociadas a cada Turno de Patrullaje.

Paquete Control de Procedimientos:

En este paquete estarán agrupados los CU relacionados con el control de los procedimientos realizados por los funcionarios durante el servicio de patrullaje: Actuaciones Policiales y Revisiones de Medio.

2.3.2 Clases arquitectónicamente significativas.

Las clases mostradas a continuación forman parte del marco arquitectónico desarrollado para soportar las funcionalidades del sistema. Cada una es la base de una jerarquía de clases pertenecientes al grupo que representan (Ramírez, 2011):

BaseBean: Constituye la raíz de la jerarquía de beans de respaldo de JSF. Proporciona muchas funcionalidades necesarias para la capa de presentación, como formato y visualización de mensaje, entre otras.

ComunFacade: Constituye la raíz de la jerarquía de las fachadas de todos los módulos. Proporciona muchas funcionalidades necesarias para la capa de presentación, como funcionalidad CRUD y funciones para consultas básicas.

EntidadPersistenteBase: Esta clase determina la raíz de la jerarquía de Entidades que se encuentra en el paquete **domain**.



Nomenclador: Esta clase representa todos los nomencladores del sistema. Unifica todos los conceptos del sistema que puedan modelarse como un nomenclador.

DaoGenerico: Esta clase es la raíz de la jerarquía de DAOs de la aplicación. Contiene numerosas funcionalidades, incluyendo soporte para todas las funcionalidades CRUD, soporte para el API simplificado de consulta, así como métodos auxiliares de uso obligatorio que determinan la forma que deben tener las Criterias, HQL y SQL de la aplicación. También soportan el paginado y constituye el punto único de acceso a las sesiones de Hibernate y la fuente de datos configurada para la aplicación.

2.3.3 Diagrama de clases del diseño

2.3.3.1 Patrones GRASP

Para la confección de los diagramas de clases del diseño se utilizaron patrones como:

Patrón Alta Cohesión: más que un diseño directamente implementable en código, se trata de un principio director que guiará en el diseño, es un objetivo subyacente a tener en cuenta continuamente. Se puede medir el nivel de cohesión en una clase cuanto más enfocado sea su comportamiento. Cada elemento del diseño debe realizar una labor única dentro del sistema, que no se desempeñe en ninguno de los demás elementos del mismo. Por otra parte pensar en interfaces fuerza a que los sistemas sigan los principios de alta cohesión.

Patrón Bajo Acoplamiento: le da respuesta a la problemática de soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización. El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento con bajo (o débil) acoplamiento no depende de demasiados otros elementos. Estos elementos pueden ser clases, subsistemas, entre otros, de ahí la importancia que se lleve a cabo el desempeño de este patrón, para de esta manera obtener una aplicación lo más flexible posible.

2.3.3.2 Diagramas de clases general

Para un mejor entendimiento de los diagramas de clases de diseño desarrollados, estos se realizaron por paquetes y divididos por capas permitiendo ganar en claridad y sencillez en los diagramas presentados.

Ambos paquetes, Organización de Patrullaje y Control de Procedimientos, comparten la misma capa de lógica de negocio y de acceso a datos, las cuales se muestran a continuación.

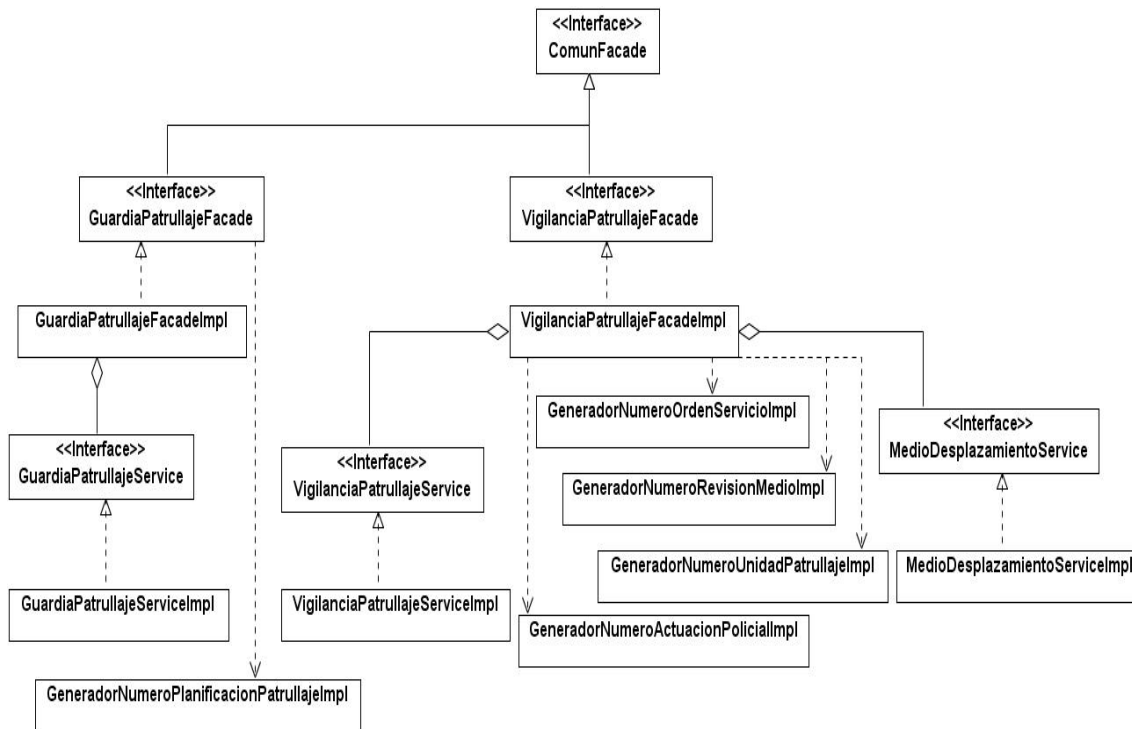


Fig. 2 Diagrama de clases. Capa de Lógica de Negocio.

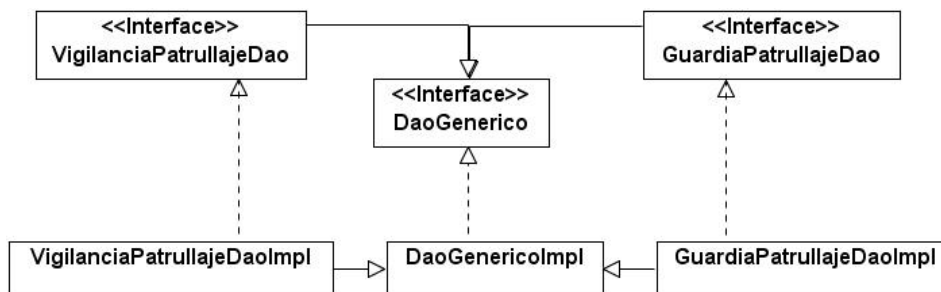


Fig. 3 Diagrama de clases. Capa de Acceso a Datos



2.3.3.3 Paquete Organización de Patrullaje

Capa de Presentación

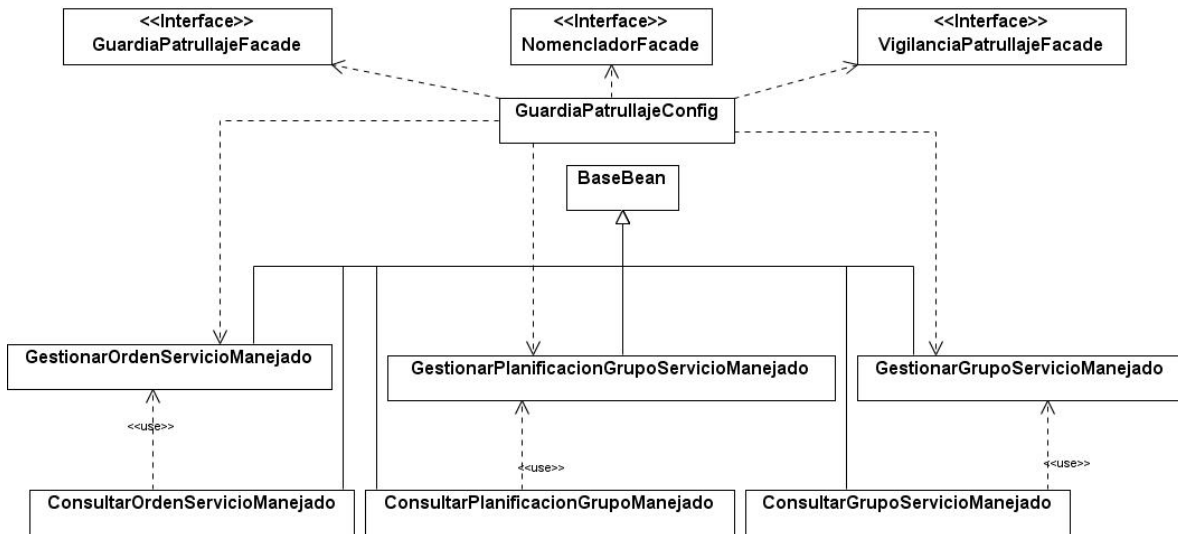


Fig. 4 Diagrama de clases. Capa de Presentación.

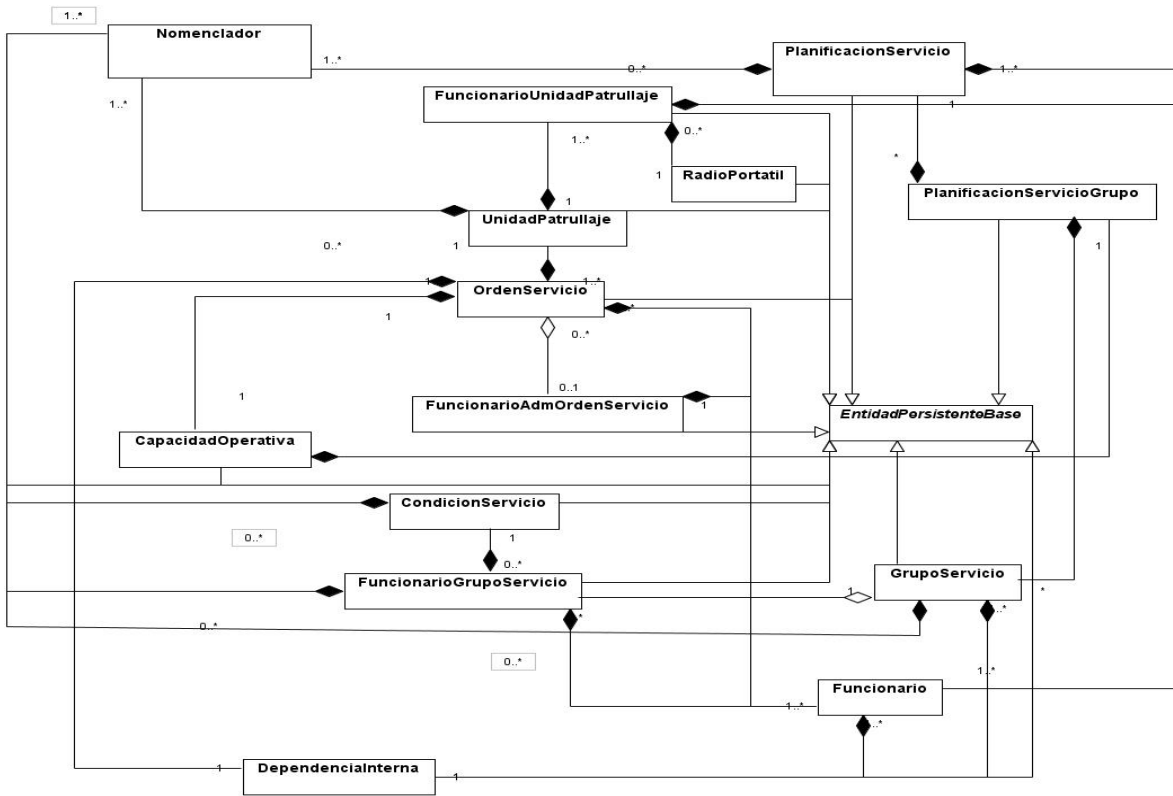


Fig. 5 Diagrama de Clases del dominio



2.3.3.4 Paquete Control de Procedimientos

Capa de Presentación

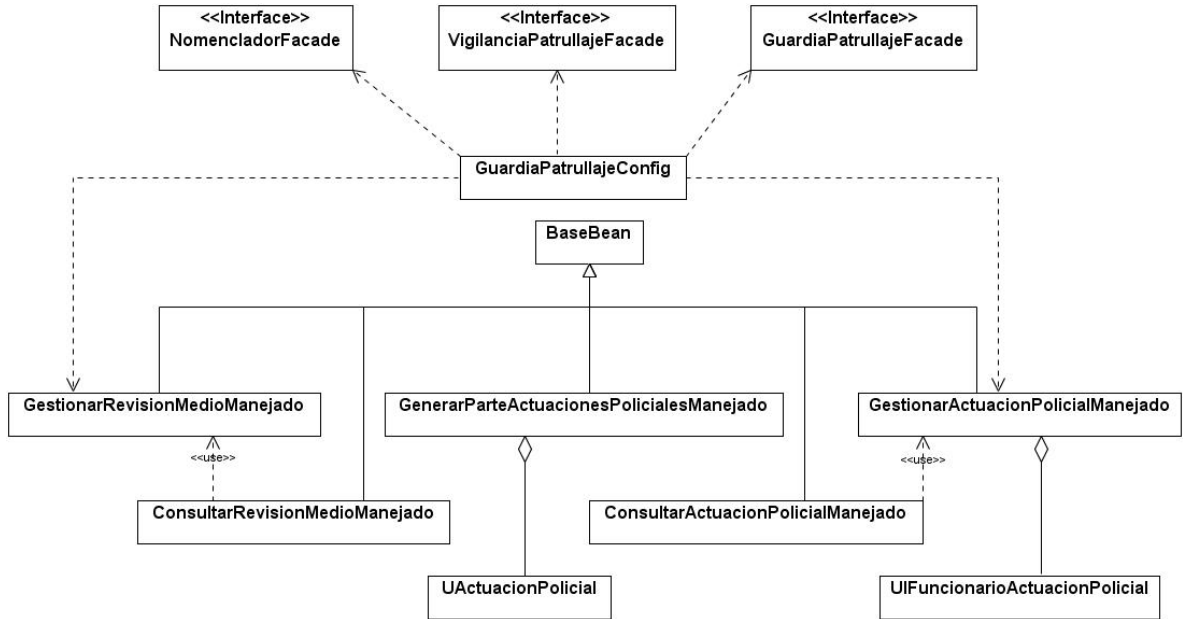


Fig. 6 Diagrama de clases. Capa de Presentación.

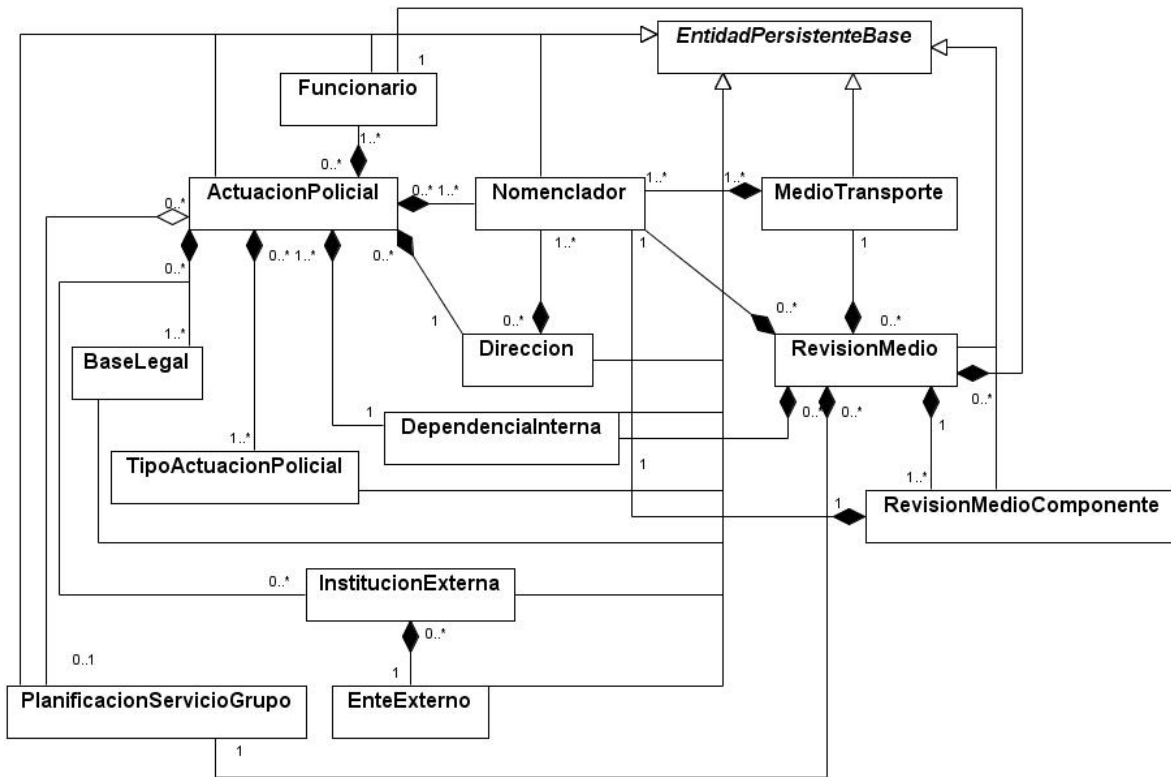


Fig. 7 Diagrama de clases del dominio

2.3.4 Descripción de clases significativas

A continuación se describen algunas de las clases más importantes para el diseño de la aplicación.

Nombre	VigilanciaPatrullajeFacadelImpl
Tipo de Clase	Fachada entre la capa de presentación y la de acceso a datos.
Atributo	Tipo
vigilanciaPatrullajeService	VigilanciaPatrullajeService
vigilanciaPatrullajeDao	VigilanciaPatrullajeDao
revisonMedioGeneradorNumero	GeneradorNumeroRevisonMedioImpl
ordenServicioGeneradorNumero	GeneradorNumeroOrdenServicioImpl
actuacionPolicialGeneradorNumero	GeneradorNumeroActuacionPolicialImpl
medioDesplazamientoService	MedioDesplazamientoService



Responsabilidad	
Nombre de la Funcionalidad	Descripción
consultarActuacionPolicial(Date desde, Date hasta, Nomenclador tipoServicio, GrupoServicio grupo, String credencial, Nomenclador estado, Nomenclador municipio, Nomenclador parroquia, Nomenclador area, Nomenclador sector, Nomenclador comunidad)	Permite consultar las Actuaciones Policiales a partir de los parámetros especificados.
consultarOrdenServicio(Date fechaInicio, Date fechaFin, Nomenclador tipo, GrupoServicio grupo, String credencial)	Permite consultar las Órdenes de Servicio a partir de los parámetros especificados.
consultarMedioTransporte(Nomenclador tipoMedio, String dependencia)	Permite consultar los Medios de Transporte a partir de los parámetros especificados.
consultarActuacionParte(Date fecha, PlanificacionServicioGrupo planificacion, Nomenclador tipoServicio, String discriminador)	Permite consultar las Actuaciones Policiales para realizar el parte de actuaciones.
salvarRevisionMedio(RevisionMedio revision)	Permite salvar una Revisión de Medio en la base de datos.
consultarRevisionMedios(Date fecha, Nomenclador tipoServicio, PlanificacionServicioGrupo turno, Nomenclador tipoMedio, String numero)	Permite consultar las Revisiones de Medio a partir de los parámetros especificados.
salvarOrdenServicio(OrdenServicio orden)	Permite salvar una Orden de Servicios en la base de datos.
salvarActuacionPolicial(ActuacionPolicial actuacion)	Permite salvar una Actuación Policial en la base de datos.
obtenerMedio(String matricula)	Permite obtener un Medio de Transporte a partir de su matrícula.

Tabla 1 Descripción de la Fachada VigilanciaPatrullajeFacadelmpl



Nombre	GuardiaPatrullajeFacadeImpl	
Tipo de Clase	Fachada entre la capa de presentación y la de acceso a datos.	
Atributo	Tipo	
grupoServicioDao	GuardiaPatrullajeDao	
funcionarioFacade	FuncionarioFacade	
generadorNumeroPlanificacionPatrullajeImpl	GeneradorNumeroPlanificacionPatrullajeImpl	
correspondenciaFacade	CorrespondenciaFacade	
Responsabilidad		
Nombre de la Funcionalidad		Descripción
consultarGrupoServicio(String identificador, String nombre, Nomenclador tipoServicio, String credencial, String discriminador)		Permite consultar los Grupos de Patrullaje a partir de los parámetros especificados.
listarGruposServicio(Integer tipo, String discriminador)		Permite obtener un listado de Grupos de Patrullaje a partir de los parámetros especificados.
validarFuncionario(String credencial)		Permite validar un Funcionario a partir de su credencial.
obtenerPlanificacionPatrullaje(Date fecha, Integer tipoServicio)		Permite obtener una Planificación de Patrullaje a partir de los parámetros especificados.
salvarActualizarPlanificacionServicio(PlanificacionServicio p, Boolean modificando)		Permite salvar una nueva Planificación de Patrullaje o modificar una existente.
consultarPlanificacionServicio(Date fechaInicio, Date fechaFin, Date diaGuardia, GrupoServicio grupoGuardia, GrupoServicio grupoPatrullaje)		Permite consultar las Planificaciones de Patrullaje a partir de los parámetros especificados.
consultarPlanificacionServicioGrupo(Date fecha, Nomenclador tipoServicio)		Permite consultar las planificaciones de patrullaje dada una fecha y un tipo de servicio

Tabla 2 Descripción de la Fachada GuardiaPatrullajeFacadeImpl



Nombre	GestionarOrdenServicioManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
planificacionPatrullaje	PlanificacionPatrullaje	
planificacionPatrullajeGrupo	PlanificacionPatrullajeGrupo	
ordenServicio	OrdenServicio	
listadoDependenciasApoyo	List<ApoyoPatrullaje>	
apoyoPatrullaje	ApoyoPatrullaje	
capacidadOperativa	CapacidadOperativa	
unidadPatrullaje	UnidadPatrullaje	
funcionarioUnidadPatrullaje	FuncionarioUnidadPatrullaje	
turnosPatrullaje	List<SelectItem>	
listadoDependencias	List<SelectItem>	
listadoFuncionariosEnGrupo	List<SelectItem>	
mediosDesplazamiento	List<SelectItem>	
funcionariosAdministrativos	List<SelectItem>	
areas	List<Nomenclador>	
funcionarioAdmOrdenServicio	FuncionarioAdmOrdenServicio	
tipoServicio	Integer	
tipoServicioObj	Nomenclador	
modificar	boolean	
Responsabilidad		
Nombre de la Funcionalidad		Descripción
agregarFuncionarioUnidad(ActionEvent e)		Permite agregar un funcionario a una unidad de patrullaje.
agregarUnidad(ActionEvent e)		Permite agregar la unidad de patrullaje.
incluir()		Se crea la orden de servicio y se



	inserta en la base de datos.
agregarFuncionarioAdministrativo(:ActionEvent e)	Permite agregar funcionarios administrativos a la orden de servicio.
agregarApoyoPatrullaje()	Permite agregar la cantidad de funcionarios que prestarán ayuda a dicha orden de servicio por Dependencia.
existeSupervisorGeneralUnidades()	Comprueba la existencia de un funcionario que tenga el cargo de Supervisor General en la Unidad de Patrullaje creada.
existeSupervisorEscuadra(Nomenclador area)	Comprueba la existencia de un funcionario que tenga el cargo de Supervisor de Escuadra para un Área determinada.
cargarDatosCapacidad()	Permite cargar los datos de la capacidad operativa de la orden de servicio.

Tabla 3 Descripción del Bean Manejado GestionarOrdenServicioManejado

Nombre	ConsultarOrdenServicioManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
tipo	Nomenclador	
nombre	String	
credencial	String	
avanzada	boolean	
fechaInicio	Date	



fechaFin	Date
grupo	GrupoPatrullaje
listaSelectGrupos	List<SelectItem>
listadoOrdenServicio	List<OrdenServicio>
Responsabilidad	
Nombre de la Funcionalidad	Descripción
consultar()	Realiza una búsqueda a partir de los criterios seleccionados por el usuario
nuevaBusqueda()	Permite limpiar los campos para realizar una nueva búsqueda.
verServicio(ActionEvent e)	Permite ver los datos de un Grupo de Patrullaje.

Tabla 4 Descripción del Bean Manejado ConsultarOrdenServicioManejado

Nombre	GestionarRevisionMedioManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
revisionMedio	RevisionMedio	
revisionMedioComponente	RevisionMedioComponente	
turnos	List<SelectItem>	
vehiculos	List<SelectItem>	
tipoMedio	Integer	
modificar	boolean	
listadoComponentes	List<RevisionMedioComponente>	
medio	Nomenclador	
Responsabilidad		
Nombre de la Funcionalidad	Descripción	
incluir()	Permite incluir la Revisión de Medio a la Base de Datos	



irAModificar()	Vuelve a cargar los datos de la revisión para así poder modificarlos
cargarTurnos()	Carga los turnos de patrullaje para seleccionar en el que se realizó la revisión

Tabla 5 Descripción del Bean Manejado GestionarRevisiónMedioManejado

Nombre	ConsultarRevisiónMedioManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
fecha	Date	
listaTurnos	List<SelectItem>	
listaNumeroMedios	List<SelectItem>	
avanzada	boolean	
listaMedios	List<MedioTransporte>	
planificacionPatrullaje	PlanificacionPatrullajeGrupo	
medioTransporte	MedioTransporte	
revisiónMedio	RevisiónMedio	
matricula	String	
tipoMedio	Nomenclador	
listadoRevisiónMedio	List<RevisiónMedio>	
Responsabilidad		
Nombre de la Funcionalidad	Descripción	
consultar()	Realiza una búsqueda a partir de los criterios seleccionados por el usuario	
nuevaBusqueda()	Permite limpiar los campos para realizar una nueva búsqueda.	
cerrar()	Lleva al usuario a la página de inicio de la aplicación.	

Tabla 6 Descripción del Bean Manejado ConsultarRevisiónMedioManejado



Nombre	GestionarGrupoServicioManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
grupo	GrupoServicio	
condicionServicio	CondicionServicio	
funcionarioGrupoPatrullaje	FuncionarioGrupoServicio	
funcionarios	List<FuncionarioGrupoServicio>	
seleccionado	Funcionario	
funcionarioVer	FuncionarioGrupoServicio	
credencial	String	
funcionarioValidado	boolean	
actualizarFuncionario	boolean	
agregarFuncValidado	boolean	
planificacionPatrullajeGrupo	PlanificacionPatrullajeGrupo	
funcionarioReposo	boolean	
funcionarioComision	boolean	
funcionarioVacaciones	boolean	
funcionarioPermiso	boolean	
funcionarioPermisoExtraordinario	boolean	
Responsabilidad		
Nombre de la Funcionalidad	Descripción	
incluirGrupoPatrullaje()	Se crea un nuevo grupo de patrullaje y se incluye en la Base de Datos.	
validarFuncionario()	Valida la existencia del funcionario para el grupo de patrullaje	
agregarFuncionario()	Añade un funcionario al grupo con su respectiva condición de servicio.	
actualizarFuncionario()	Carga los datos del funcionario en el grupo y permite	



	cambiarle la condición de servicio
--	------------------------------------

Tabla 7 Descripción del Bean Manejado GestionarGrupoServicioManejado

Nombre	ConsultarGrupoServicioManejado	
Tipo de Clase	Bean de Respaldo	
	Atributo	Tipo
	tipo	Nomenclador
	listadoGrupos	List<GrupoServicio>
	nombre	String
	credencial	String
	identificador	String
	Responsabilidad	
	Nombre de la Funcionalidad	Descripción
	consultar()	Realiza una búsqueda a partir de los criterios seleccionados por el usuario
	nuevaBusqueda()	Permite limpiar los campos para realizar una nueva búsqueda.
	cerrar()	Lleva al usuario a la página de inicio de la aplicación..
	verServicio()	Permite ver los datos de un grupo de patrullaje seleccionado

Tabla 8 Descripción del Bean Manejado ConsultarGrupoServicioManejado

Nombre	GestionarActuacionPolicialManejado	
Tipo de Clase	Bean de Respaldo	
	Atributo	Tipo
	actuacionPolicial	ActuacionPolicial
	listaTurnos	List<SelectItem>



gestionarBaseLegalSubviewManejado	GestionarBaseLegalSubviewManejado
asociado	String
areas	List<Nomenclador>
credencial	String
seleccionado	Funcionario
listadoFuncionarios	List<UIFuncionarioActuacionPolicial>
institucionExterna	InstitucionExterna
tipoActuacionPolicial	TipoActuacionPolicial
tipoServicio	Integer
modificar	boolean
listaTipoActuacion	List<TipoActuacionPolicial>
listaInstitucionExterna	List<InstitucionExterna>
Responsabilidad	
Nombre de la Funcionalidad	Descripción
agregarFuncionario(ActionEvent e)	Funcionalidad que permite adicionar funcionarios a la actuación
agregarInstitucionExterna(ActionEvent e)	Permite adicionar las instituciones externas que participaron en la actuación y una descripción sobre ellas.
validarFuncionario(ActionEvent e)	Valida y selecciona un funcionario, si existe, para asociarlo a la actuación policial.
incluir()	Incluye la actuación policial a la Base de Datos.

Tabla 9 Descripción del Bean Mandejado GestionarActuacionPolicialManejado

Nombre	ConsultarActuacionPolicialManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	



listadoActuaciones	List<ActuacionPolicial>
fechaDesde	Date
fechaHasta	Date
tipoServicio	Nomenclador
grupoPatrullaje	GrupoServicio
estado	Nomenclador
municipio	Nomenclador
parroquia	Nomenclador
area	Nomenclador
sector	Nomenclador
comunidad	Nomenclador
credencial	String
avanzada	Boolean
listadoGrupos	List<SelectItem>
Responsabilidad	
Nombre de la Funcionalidad	Descripción
consultar()	Devuelve el resultado de la búsqueda
nuevaBusqueda()	Limpia los campos para que el usuario pueda insertar nuevos valores para los criterios de búsqueda
cerrar():	Cierra la página y lleva al usuario a la página principal de la aplicación.

Tabla 10 Descripción del Bean Manejado ConsultarActuacionPolicialManejado

Nombre	GenerarParteActuacionesPolicialesManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
fecha	Date	



planificacionPatrullaje	PlanificacionPatrullajeGrupo
tipoServicio	Nomenclador
listaTurnos	List<SelectItem>
listadoActuaciones	List<ActuacionPolicial>
cantidadActuaciones	int
faltas	int
delitos	int
serviciosOrdinarios	int
serviciosEspeciales	int
colaboraciones	int
actuacionesArea	List<UActuacionPolicial>
actuacionesTipoAct	List<UActuacionPolicial>
Responsabilidad	
Nombre de la Funcionalidad	Descripción
cargarTurnos()	Carga los turnos de patrullaje en dependencia de la fecha y el tipo de servicio seleccionado por el usuario.
consultar()	Realiza la consulta con la Base de Datos para obtener las actuaciones policiales realizadas en dependencia de los criterios introducidos por el usuario.
actualizarCategorias()	Calcula las cantidades de actuaciones policiales realizadas para cada Categoría de Actuación Policial.
actuacionesPorTipoAct()	Calcula las cantidades de actuaciones policiales realizadas para cada Tipo de Actuación Policial.
actuacionesPorArea()	Calcula las cantidades de actuaciones policiales realizadas por áreas.

Tabla 11 Descripción del Bean Manejado GenerarParteActuacionesPolicialesManejado



Nombre	GestionarPlanificacionGrupoServicioManejado	
Tipo de Clase	Bean de Respaldo	
Atributo	Tipo	
planificacionPatrullaje	PlanificacionPatrullaje	
listadoGrupoPatrullaje	List<SelectItem>	
planificacionPatrullajeGrupo	PlanificacionPatrullajeGrupo	
modificar	boolean	
tipoServicio	Integer	
planificaciones	List<PlanificacionPatrullajeGrupo>	
Responsabilidad		
Nombre de la Funcionalidad	Descripción	
incluirPlanificacion()	Permite incluir o actualizar una planificación de patrullaje.	
actualizarGrupos()	Permite actualizar el listado de grupos a seleccionar por el usuario para realizarle la planificación.	

Tabla 12 Descripción del Bean Manejado GestionarPlanificaciónGrupoServicioManejado

2.4 Modelo de Base de Datos

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos. Un modelo de datos describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación. Con el objetivo de lograr la persistencia de los datos y que estos puedan ser consultados en cualquier momento por cualquier usuario se utilizó, en la concepción de este prototipo, una base de datos relacional. El uso de esta técnica de almacenamiento está muy difundido actualmente debido a las garantías que ofrece en cuanto a la durabilidad de los datos se refiere (Pressman).

2.4.1 Diagrama de clases persistentes



Las clases persistentes representan almacenamientos de datos que persistirán más allá de la ejecución del software. La persistencia en la aplicación se maneja con la ayuda del framework Hibernate para el cual es necesario realizar un mapeo de cada una de las entidades persistentes con cada una de las tablas correspondientes en la base de datos. A continuación se presenta el diagrama de clases que se determinaron como persistentes en cada uno de los paquetes.

2.4.1.1 Paquete Organización de Patrullaje

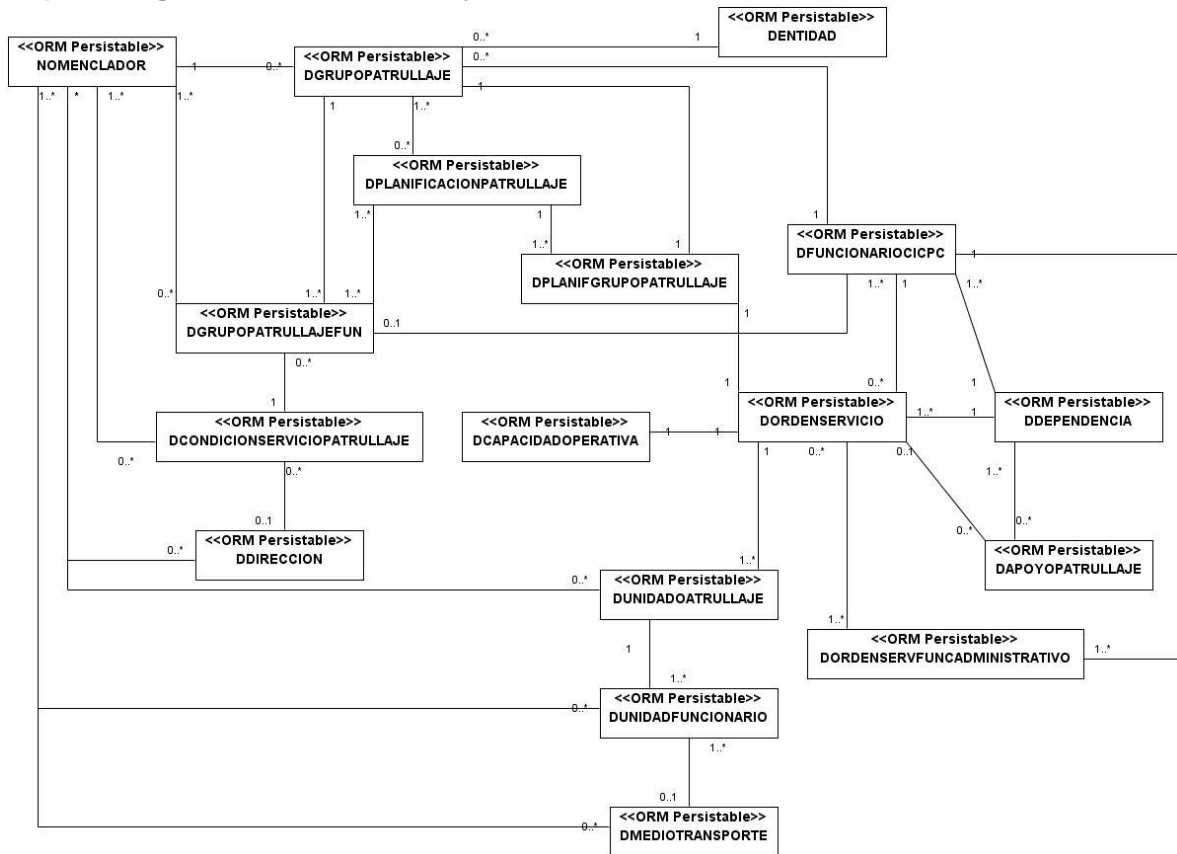


Fig. 8 Diagramas de Clases Persistentes. Paquete Organización de Patrullaje.



2.4.1.2 Paquete Control de Procedimientos

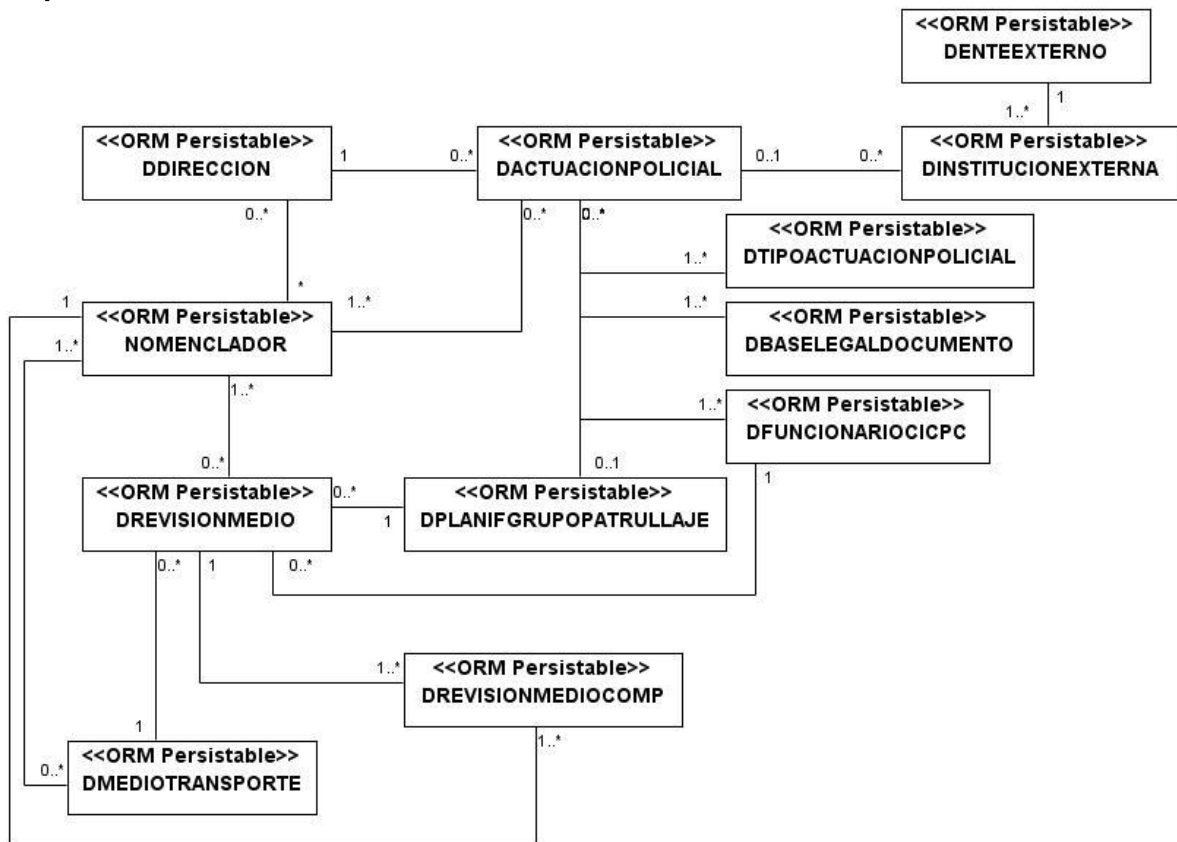


Fig. 9 Diagramas de Clases Persistentes. Paquete Control de Procedimientos.

2.4.2 Diseño de la Base de Datos

El diseño de la Base de Datos se realiza con el objetivo de garantizar que los datos persistentes se almacenen de forma coherente y eficaz. Así como para definir el comportamiento que se debe implementar en la base de datos. A continuación se presenta el diseño de la Base de Datos definidos en cada uno de los paquetes (Pressman).



2.4.2.2 Paquete Organización de Patrullaje

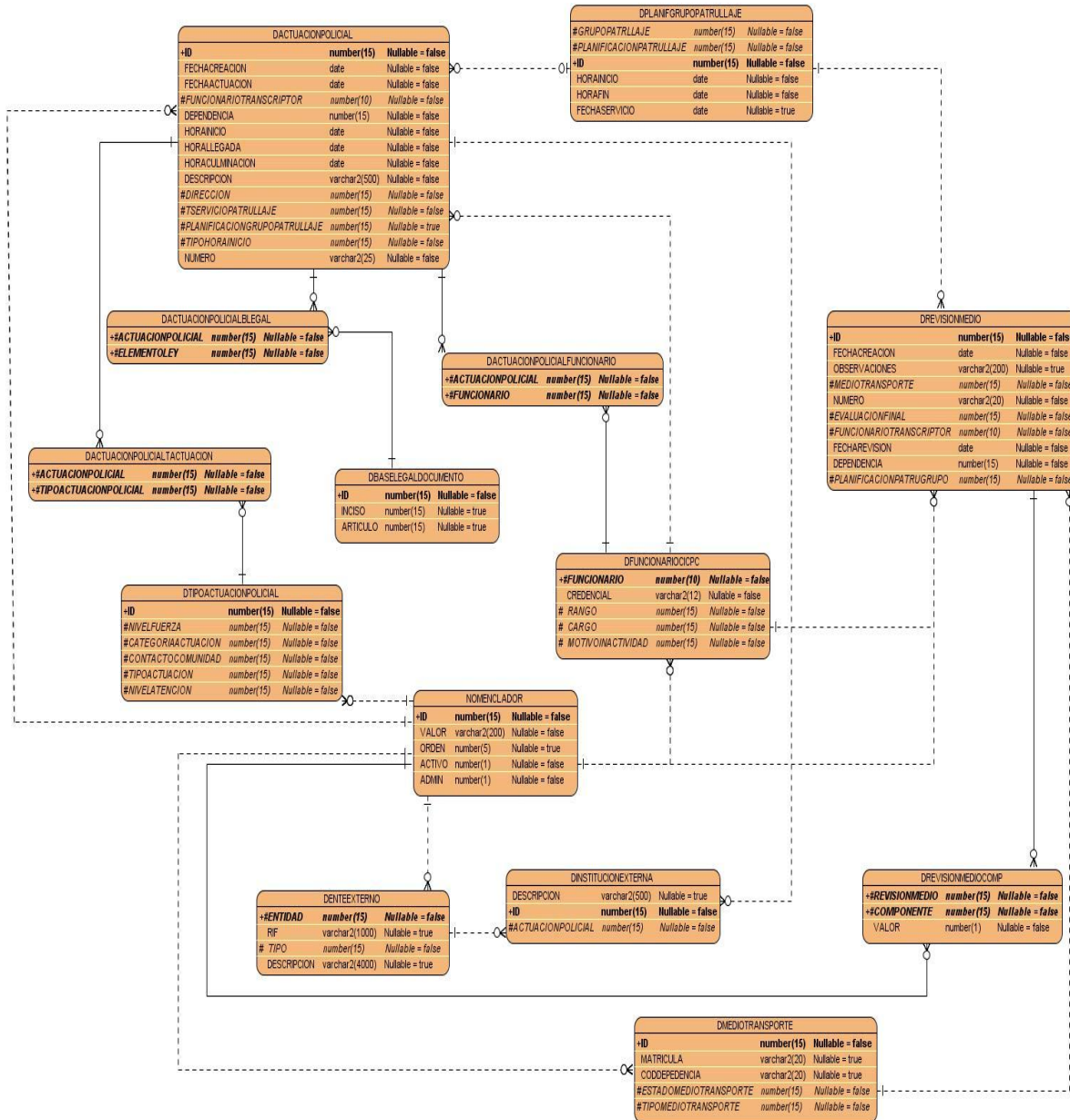


Fig. 11 Diseño de la Base de Datos. Paquete Organización de Patrullaje



2.5 Conclusiones

Se realizó el diseño del Subsistema Vigilancia y Patrullaje de la Personalización y Ampliación del SIIPOL definiendo las clases que se utilizarán para la posterior implementación del sistema y las relaciones entre las mismas, teniendo en cuenta principalmente las facilidades que los framework aportaban y los patrones de diseño que sin forzar al programador ayudan a la solución propuesta. También se realizó el modelo de datos que se necesitará para el almacenamiento de la información. Todo esto con el objetivo de satisfacer los requerimientos que necesita este sistema y que pueda desarrollarse de forma cómoda el modelo de implementación de la solución propuesta, ya que proporcionan artefactos necesarios como son los diagramas de clases del diseño, diagrama de clases del dominio.



CAPÍTULO 3. Implementación y Prueba de la solución propuesta.

3.1 Introducción

Una vez diseñada la propuesta de solución se centran los esfuerzos en la implementación de la misma. Se presentarán los diagramas de componentes del módulo. Además, se verificará el resultado de la implementación, para ello se hará uso de los artefactos que fueron generados durante el trabajo de pruebas de esta forma quedarán expuestos los resultados obtenidos por las pruebas realizadas al módulo.

3.2 Diagramas de componentes

Los diagramas de componentes muestran las relaciones de dependencia entre las partes modulares de un sistema y encapsulan la implementación. Se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias lógicas entre un conjunto de componentes de software, sean estos componentes de código fuente, librerías, binarios o ejecutables. (Jacobson, y otros, 2007)

Para una mejor organización en el módulo se ha realizado un diagrama de componentes por cada subsistema del módulo.

3.2.1 Diagrama de componentes generales del módulo

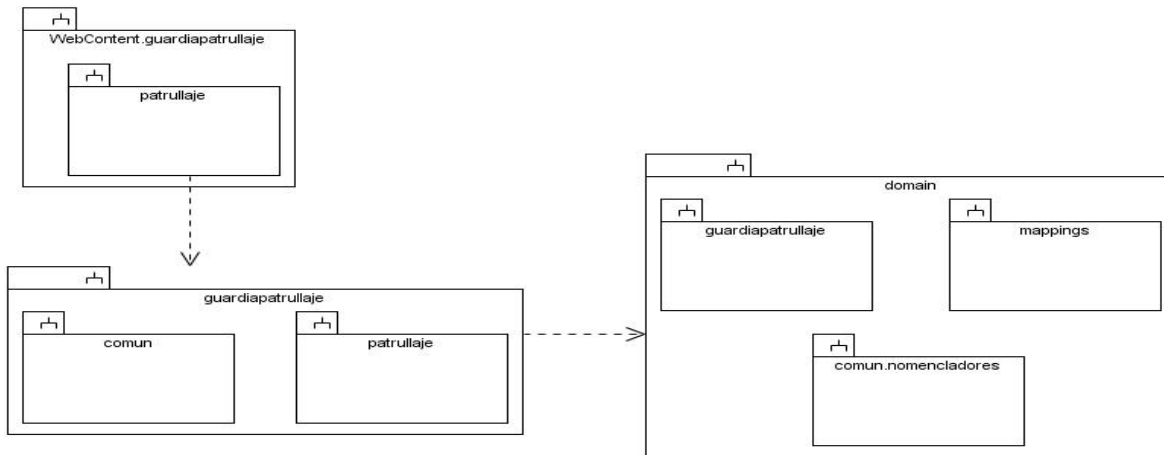


Fig. 12 Diagrama de componentes generales

3.2.2 Componentes del Dominio

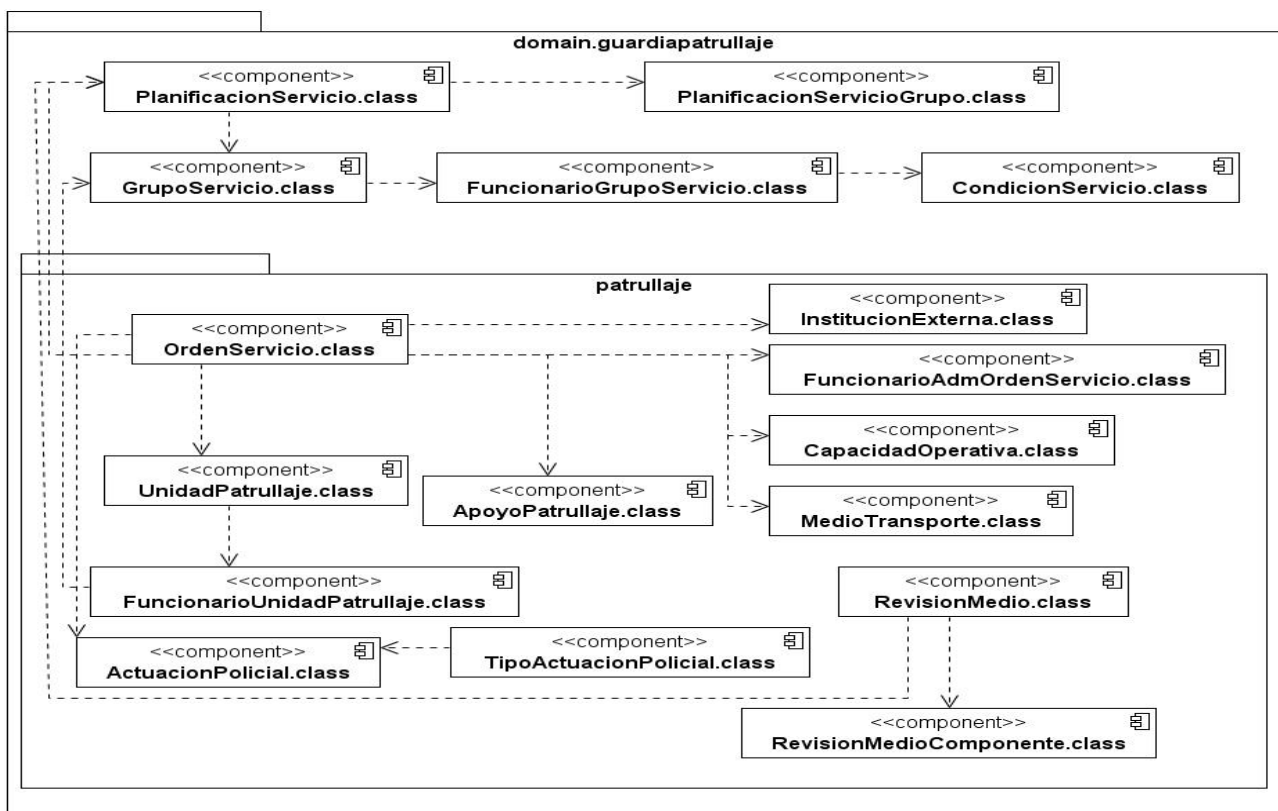


Fig. 13 Componentes del Dominio



3.2.3 Mappings

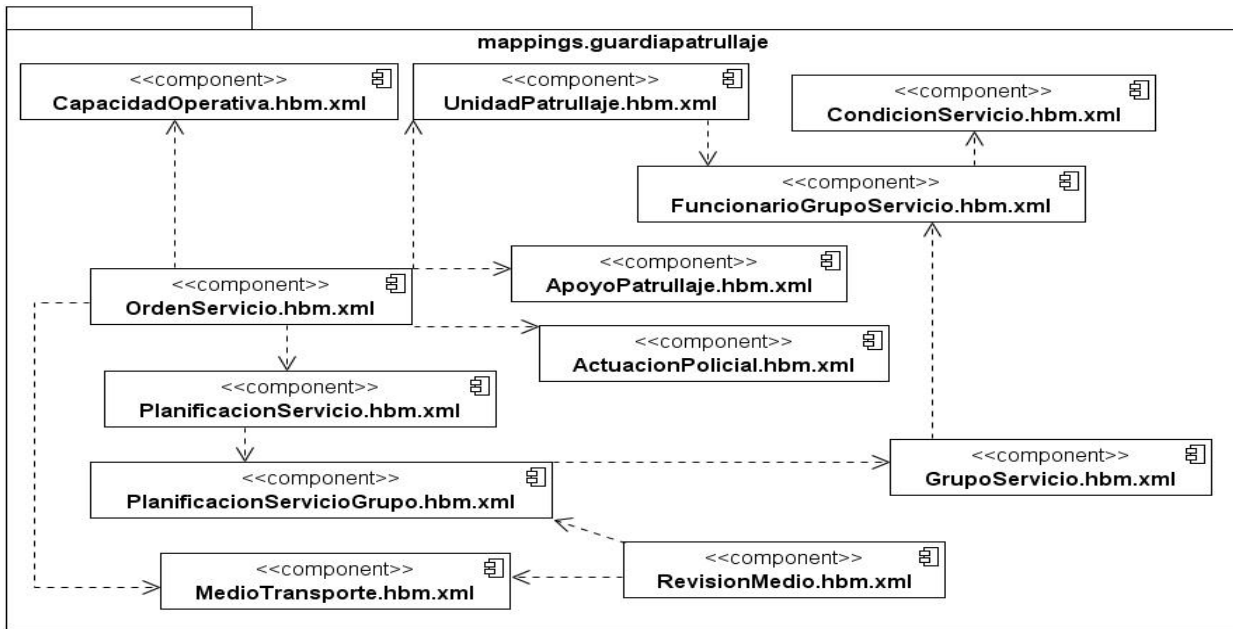


Fig. 14 Componentes Mappings

3.2.4 Archivos JSP

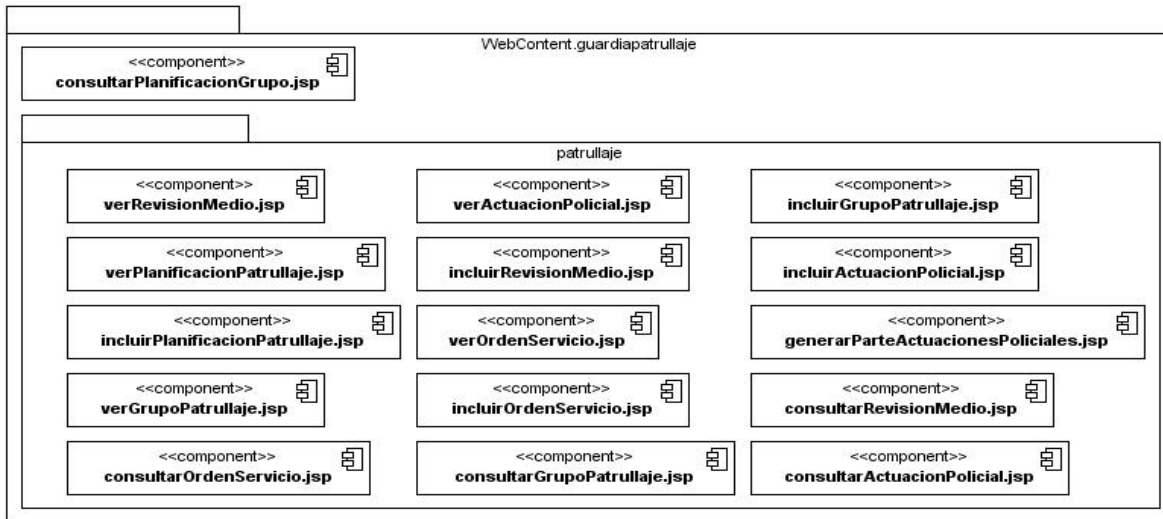


Fig. 15 Componentes JSP

3.2.5 Subsistema Común

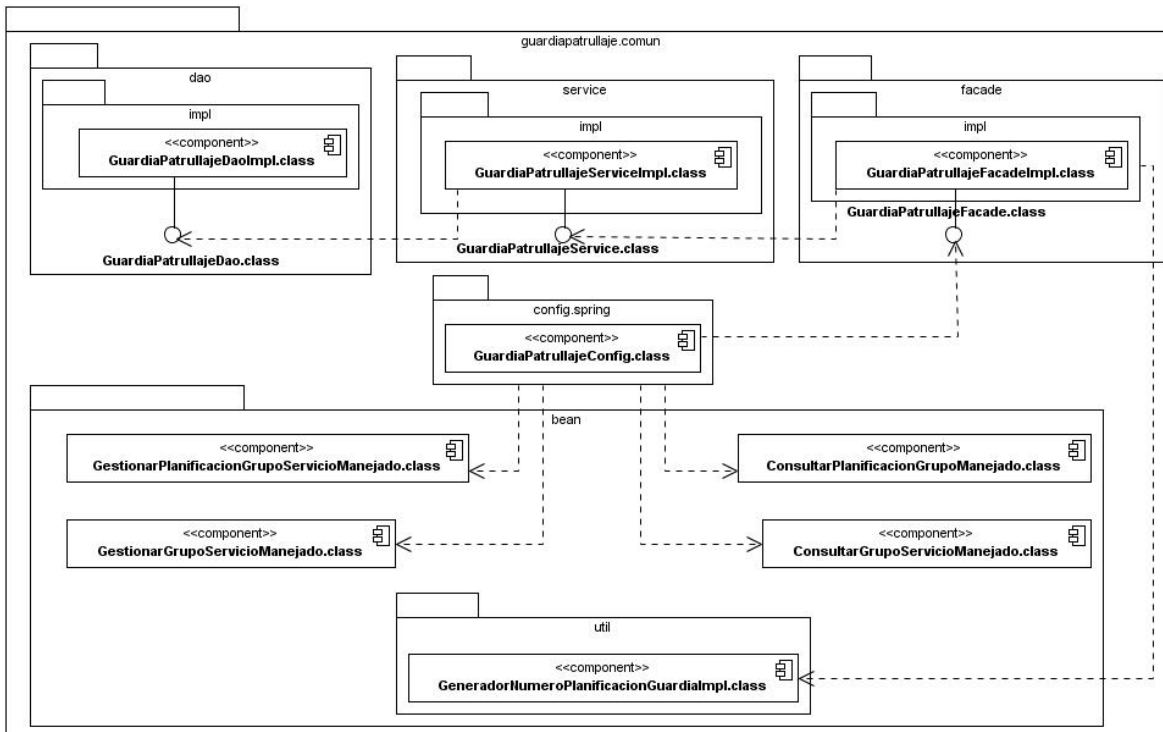


Fig. 16 Componentes Subsistema Común



3.2.6 Subsistema Patrullaje

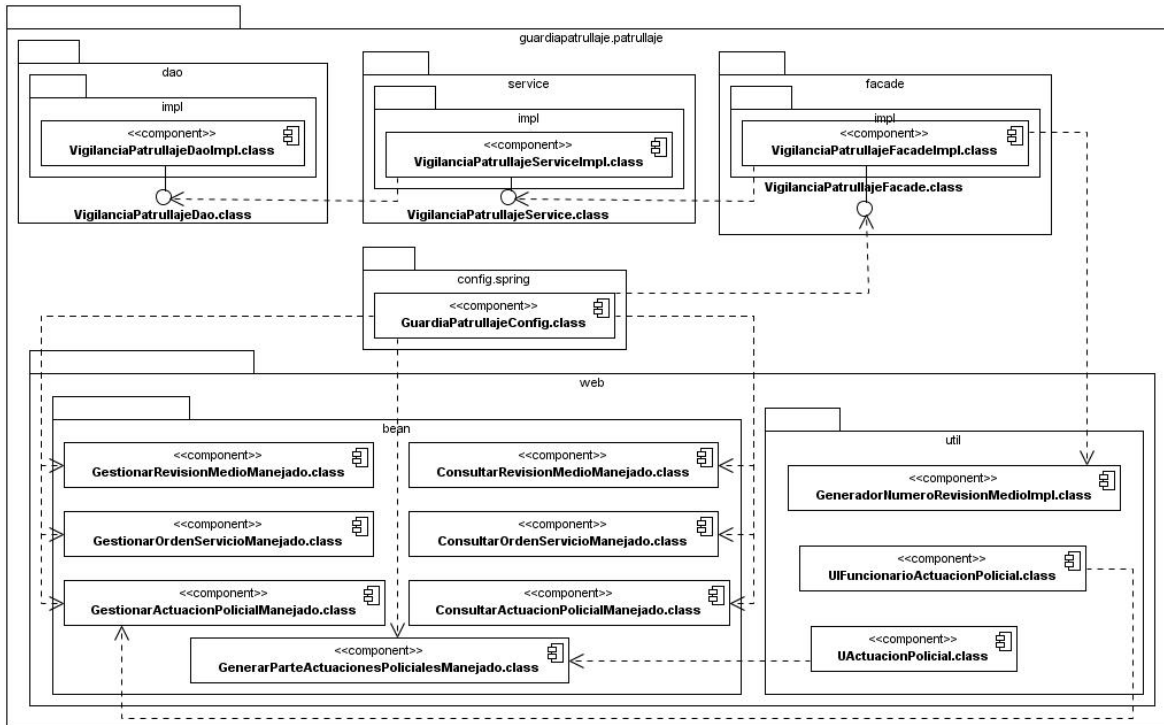


Fig. 17 Componentes Subsistema Patrullaje

3.3 Pruebas de la solución.

El proceso de pruebas en el desarrollo del software debe estar caracterizado por una buena planificación y control de las actividades que durante el mismo se realicen. Las pruebas son un conjunto de actividades que se planifican con anticipación y que deben ser realizadas de manera sistemática. Roger Pressman define las pruebas como “*el último bastión para asegurar la calidad y, de manera más pragmática, el descubrimiento de errores*” (Pressman). Es por ello que una vez completada la implementación de la aplicación es necesario probarla para poder encontrar y corregir la mayor cantidad de errores posibles antes de entregársela al cliente. Las pruebas aplicadas al módulo de Vigilancia y Patrullaje fueron pruebas de “caja negra”.



3.3.1 Estrategia de pruebas.

Para la ejecución de las pruebas se definió una estrategia con: niveles, tipos de pruebas, métodos y técnicas correspondientes, con el objetivo de buscar errores para el perfeccionamiento la de solución implementada.

Las pruebas para el módulo de realizaron en 3 bloques, cada uno con 3 iteraciones. La estrategia definida se centra en el nivel de pruebas de sistema, como tipo de prueba las de funcionalidad y dentro de estas las de función, utilizando el método de caja negra. Las pruebas de sistema son realizadas cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

3.3.2 Pruebas de Caja Negra

Las pruebas de Caja Negra son pruebas que se realizan cuando se conoce la función específica para la que se diseño el producto. Las mismas tienen como principal objetivo demostrar que cada función es plenamente operacional, a la par que se buscan los errores en la aplicación. Estas pruebas centran sus esfuerzos en los requisitos funcionales del software. Estas pruebas tratan, esencialmente, de encontrar los siguientes errores:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en estructuras de datos o en el acceso a la base de datos.
- Errores de comportamiento y desempeño.
- Errores de inicialización y término.

Para lograr encontrar la mayor cantidad de errores posibles se diseñaron casos de pruebas (CP) para cada uno de los casos de uso del módulo (Ver Anexo 1).

Las pruebas fueron divididas en dos grupos, pruebas internas y pruebas cruzadas.

Pruebas Internas: Son realizadas por el equipo de calidad interna del proyecto con el fin de entregar un producto con la menor cantidad de errores posibles. Se centran en el cumplimiento de las funcionalidades descritas en el listado de requerimientos y de casos de uso.



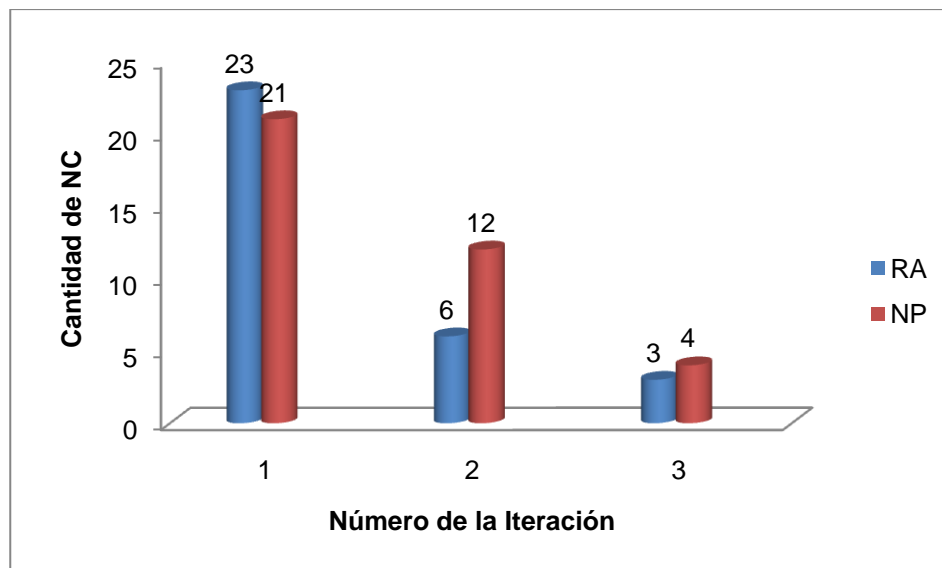
Pruebas Cruzadas: Pruebas realizadas al sistema por el resto de los equipos de desarrollo del proyecto. Se realizaron con el fin de encontrar la mayor cantidad de errores posibles en término de validaciones, pautas definidas por arquitectura de información, formato de los campos, entre otras.

En el primer grupo se probaron, por parte del equipo de calidad interna del proyecto, los casos de uso:

- CU Gestionar Grupo de Patrullaje.
- CU Gestionar Orden de Servicio.
- CU Gestionar Planificación de Patrullaje.
- CU Gestionar Actuación Policial.

Las cantidades de no conformidades resultas (RA) y las que no precedieron (NP) se muestran a continuación:

Pruebas I Etapa:



Gráfica 1 Resultados de las pruebas

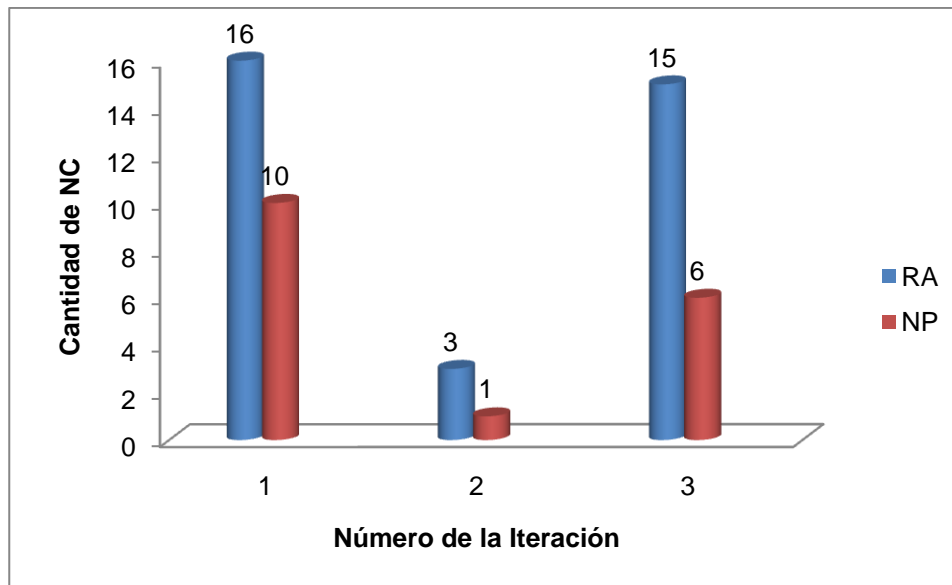
Luego de la realización de estas pruebas se le presentó al cliente lo que estaba implementado para una pre aceptación por parte del mismo. Durante esta etapa el cliente interactuó con el módulo, aparentando



un ambiente real. Como resultado de esto no fueron detectadas no conformidades y se obtuvieron cinco pedidos de cambios.

En el segundo grupo se probaron todos los CU y las mismas fueron pruebas cruzadas realizadas por otros equipos de desarrollo del proyecto. Los resultados obtenidos en estas pruebas se muestran a continuación:

Pruebas II Etapa:



Gráfica 2 Resultados de las pruebas

Esto muestra como los errores encontrados fueron corregidos entre cada una de las iteraciones, aumentando así la calidad del producto desarrollado.

3.4 Conclusiones

Se puede concluir en que, los diagramas generados en el modelo de implementación, visualizan los diferentes subsistemas de los que se compone el módulo, además muestran cada uno de los diferentes componentes que lo conforman y sus relaciones, logrando un mejor entendimiento del sistema desde el punto de vista de implementación. Además, los resultados obtenidos en las pruebas realizadas, ejemplifican la calidad del trabajo efectuado en la implementación del Módulo Vigilancia y Patrullaje.



Conclusiones Generales

Una vez concluida la investigación y la implementación del Módulo Vigilancia y Patrullaje de la personalización y ampliación del SIIPOL, se le han dado cumplimiento a los objetivos planteados y se obtuvieron los resultados que a continuación se mencionan:

- Se estructuraron los mecanismos que permiten la comunicación entre la CVP y las diferentes áreas.
- Se realizó la integración de los componentes que conforman el proceso para la elaboración de la Orden de Servicio.

Para lograr esto se realizó el diseño de la solución del Módulo Vigilancia y Patrullaje, se implementaron 11 Casos de Uso y se probaron las funcionalidades a través de 11 Casos de Prueba de Aceptación. Todo el proceso de desarrollo fue guiado por la metodología RUP.



Recomendaciones

Una vez terminado el desarrollo del módulo se recomienda:

- Continuar practicándole pruebas a las funcionalidades implementadas con el objetivo de garantizar una aplicación más estable.
- Estudiar la posibilidad de incrementar otras funcionalidades que faciliten el trabajo de los funcionarios de la Coordinación de Vigilancia y Patrullaje.



Referencias Bibliográficas

1. **Policía, Consejo General de.** *Colección BAQUIA: REGLAS MÍNIMAS DE ESTANDARIZACIÓN PARA LOS CUERPOS POLICIALES. 3. PASOS Y HUELLAS. Manual sobre Procedimientos Policiales.* Caracas, Venezuela : s.n., Mayo, 2010.
2. **Arriagada, Irma y Godoy, Lorena.** *SEGURIDAD CIUDADANA Y VIOLENCIA EN AMÉRICA LATINA.* CEPAL, Naciones Unidas. Washington : CEPAL, 2001. pág. 3.
3. **Digital, Contenido.** Contenido Digital. *Hibernate una Herramienta Mapping.* [En línea] <http://contenidodigital.wordpress.com/2008/09/19/hibernate-una-herramienta-de-mapping-ii/>.
4. **Soto, Carlos Bustos.** Centro Nacional de Estudios de Orden Público. [En línea] [Citado el: 23 de 4 de 2011.] http://www.ceneop.cl/index.php?option=com_content&view=article&id=391:orden-publico&catid=98:news&Itemid=586.
5. **Velásquez, Hugo Acero.** *Situación de violencia y delincuencia en Venezuela y concentración de delincuencia en Caracas.* Caracas, Venezuela : s.n., 2006.
6. **Definición.** Definición. [En línea] [Citado el: 12 de abril de 2011.] <http://definicion.de/policia/>.
7. Seguridad Pública. [En línea] [Citado el: 13 de junio de 2011.] <http://www.seguridadpublica.es/2008/10/el-modelo-espanol-funciones-y-competencias-de-las-fuerzas-y-cuerpos-de-seguridad-del-estado-de-las-policias-autonomicas-y-de-las-policias-locales-relaciones-entre-las-diferentes-fuerzas-y-cuerpos-d-2/>.
8. **Emanuelsson, Dick.** El Policía Democrático nace en Venezuela. *Consejo General de Policía.* [En línea] 10 de 10 de 2010. [Citado el: 05 de 11 de 2010.] <http://www.consejopolicia.gob.ve/index.php/noticias-cgp/articulos-de-opinion/698-el-policia-democratico-nace-en-venezuela>.
9. **Policía, Consejo General de.** *Sesión 2: Policía Nacional Bolivariana y vigencia del Patrullaje Policial.* 2010.
10. —. *Colección BAQUIA: REGLAS MÍNIMAS DE ESTANDARIZACIÓN PARA LOS CUERPOS POLICIALES. 6. POR LAS CALLES DEL PUEBLO. Practiguía sobre el Sistema Integrado de Información y Dirección de las Operaciones Policiales.* Caracas, Venezuela : s.n., Mayo, 2010.
11. —. *Manual sobre Procedimientos Policiales.* Caracas, Venezuela : s.n., 2010.



12. **Cohen, Daniel y Asín, Enrique.** *Sistemas de información para los negocios Un enfoque de toma de decisiones.* Mexico : Ed. McGraw Hill, 2000. pág. 4.
13. **Mapas Digitales, S.A.** Dmapas-SEGURIDAD CIUDADANA Y GESTIÓN POLICIAL. [En línea] [Citado el: 21 de 11 de 2010.] http://www.dmapas.cl/productos_stegpol.htm.
14. **Pressman, Roger S.** *INGENIERÍA DEL SOFTWARE. Un enfoque práctico.* Sexta Edición.
15. **Jacobson, Ivar, Rumbaugh, James y Booch, Grady.** *El lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Addison Wesley, 2007. Segunda Edición.
16. **Larman, Craig.** *UML y Patrones.* Mexico : Prentice Hall, 1999.
17. **Organización, Visual Paradigm.** Sitio Web oficial Visual-Paradigm. [En línea] 2009. [Citado el: 19 de 11 de 2010.] <http://www.visual-paradigm.com/product/vpum/>.
18. **Objetos, Laboratorio de Tecnología Orientada a.** Universidad de San Martín de Porres <http://www.usmp.edu.pe>. [En línea] 2002. [Citado el: 20 de 11 de 2010.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info40/j2ee.html>.
19. **Alvarez, Miguel Angel.** DesarrolloWeb.com. [En línea] 18 de 7 de 2001. [Citado el: 20 de 11 de 2010.] <http://www.desarrolloweb.com/articulos/497.php>.
20. **Oviedo, Universidad de.** Sitio Web de la E.U de Ingeniería Técnica Informática de Oviedo. [En línea] [Citado el: 14 de junio de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
21. **LaFlecha.** La flecha,tu diario de ciencia y tecnología. *Red Hat presenta "Developer Studio" basado en Eclipse.* [En línea] 27 de 8 de 2007. [Citado el: 21 de 11 de 2010.] <http://www.laflecha.net/canales/softlibre/noticias/red-hat-presenta-developer-studio-basado-en-eclipse>.
22. **Torrijos, Ricard Lou.** Programación en Castellano. *Introducción a la Tecnología JavaServer Faces.* [En línea] [Citado el: 21 de 11 de 2010.] http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233.
23. **Ramón, Óscar Sánchez.** *Java Server Faces.* [En línea] 2006. [Citado el: 20 de 11 de 2010.] <http://ditec.um.es/ssdd/trabajos/0506/exposicionJSF.ppt>.
24. **Torrijos, Ricard Lou.** Programación en Castellano. *Introducción a la Tecnología JavaServer Faces .* [En línea] 2003. [Citado el: 22 de 11 de 2010.] http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233.



25. **Piñeiro, Juan Medín y Figueras, Antonio García.** *Hacia una arquitectura con JavaServer Faces.* Sevilla : s.n., 2006.
26. **Walls, Crairg y Breidenbach, Ryan.** *SPRING IN ACTION.* Greenwich : Manning Publications Co, 2005.
27. **Ramírez, Jorge Amado Soria.** *Descripción de la Arquitectura de Software Proyecto CICPC-PNB.* 2011. SIIPOL 5.0.



Bibliografía

- Alvarez, Miguel Angel. 2001.** DesarrolloWeb.com. [En línea] 18 de 7 de 2001. [Citado el: 20 de 11 de 2010.] <http://www.desarrolloweb.com/articulos/497.php>.
- Arriagada, Irma y Godoy, Lorena. 2001.** *SEGURIDAD CIUDADANA Y VIOLENCIA EN AMÉRICA LATINA*. CEPAL, Naciones Unidas. Washington : CEPAL, 2001. pág. 3.
- Baver, Christian y King, Gavin. 2005.** *Hibernate in Action*. Greenwich : Manning Publications Co, 2005.
- Bolivariana, Policía Nacional.** Policía Nacional Bolivariana. [En línea] [Citado el: 26 de Octubre de 2010.] <http://www.policianacional.gob.ve/>.
- Cabeza, Eduardo Collado y Berenguer, Angi Díaz. 2003.** Madritel. [En línea] 2003. [Citado el: 18 de 11 de 2010.] <http://web.madritel.es/personales3/edcollado/ingsw/tema2/2-6.htm>.
- Cohen, Daniel y Asín, Enrique. 2000.** *Sistemas de información para los negocios Un enfoque de toma de decisiones*. Mexico : Ed. McGraw Hill, 2000. pág. 4.
- Copyright © 1997-2010 Object Management Group, Inc. All Rights Reserved.** UNIFIED MODELING LANGUAGE. UML® Resource Page. [En línea] [Citado el: 20 de 11 de 2010.] <http://www.uml.org>.
- Definición.** Definición. [En línea] [Citado el: 12 de abril de 2011.] <http://definicion.de/policia/>.
- Digital, Contenido.** Contenido Digital. *Hibernate una Herramienta Mapping*. [En línea] <http://contenidodigital.wordpress.com/2008/09/19/hibernate-una-herramienta-de-mapping-ii/>.
- Emanuelsson, Dick. 2010.** El Policía Democrático nace en Venezuela. *Consejo General de Policía*. [En línea] 10 de 10 de 2010. [Citado el: 05 de 11 de 2010.] <http://www.consejopolicia.gob.ve/index.php/noticias-cgp/articulos-de-opinion/698-el-policia-democratico-nace-en-venezuela>.
- Figuroa, Roberth G. y Solís, Camilo J. 2007.** *Metodologías Tradicionales Vs. Metodologías Ágiles*. 2007.



- García, Dexty. 2008.** Venezuela tendrá un nuevo sistema de seguridad ciudadana . *MinCI*. [En línea] 8 de 3 de 2008. [Citado el: 05 de 11 de 2010.]
http://www.minci.gob.ve/pagina/2/175271/venezuela_tendrun_nuevo.html.
- Guatemala, PNUD. 1998.** *Guatemala: Los contrastes del desarrollo humano*. 1998.
- Gutiérrez, Javier. 2009.** *Una comparación entre C, C++, Java y Ada*. Cantabria : s.n., 2009.
- Jacobson, Ivar, Rumbaugh, James y Booch, Grady. 2007.** *El lenguaje Unificado de Modelado. Manual de Referencia*. Madrid : Addison Wesley, 2007. Segunda Edición.
- . **2007.** *El lenguaje Unificado de Modelado. Manual de Referencia*. Segunda Edición. Madrid : Addison Wesley, 2007.
- . **2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educacion, 2000. 120.
- Juan Alonso Ramos.** Introducción a Ajax4jsf. *Adictos al Trabajo*. [En línea] [Citado el: 4 de 12 de 2010.]
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf..>
- Justicia, Ministerio del Poder Popular para las Relaciones de Interior y.** Ministerio del Poder Popular para las Relaciones de Interior y Justicia. [En línea] [Citado el: 27 de Octubre de 2010.]
<http://www.mpprij.gob.ve>.
- LaFlecha. 2007.** La flecha,tu diario de ciencia y tecnologia. *Red Hat presenta "Developer Studio" basado en Eclipse*. [En línea] 27 de 8 de 2007. [Citado el: 21 de 11 de 2010.]
<http://www.laflecha.net/canales/softlibre/noticias/red-hat-presenta-developer-studio-basado-en-eclipse>.
- Larman, Craig. 1999.** *UML y Patrones*. Mexico : Prentice Hall, 1999.
- Letelier, Patricio. 2003.** *Proyecto Docente e Investigador*. 2003.
- LuAuF. 2008.** LuAuF. *Entornos de Desarrollo Integrado para Java*. [En línea] 2008. [Citado el: 20 de 11 de 2010.] <http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java>.
- Mapas Digitales, S.A.** Dmapas-SEGURIDAD CIUDADANA Y GESTIÓN POLICIAL. [En línea] [Citado el: 21 de 11 de 2010.] http://www.dmapas.cl/productos_stegpol.htm.
- Montenegro, Ximena. 1997.** *GUIA PARA LA IDENTIFICACION Y FORMULACION DE PROYECTOS DE MEJORAMIENTO DE BARRIOS EN EXTREMA POBREZA*. Santiago de Chile : ILPES, 1997.
- Objetos, Laboratorio de Tecnología Orientada a.** 2002. Universidad de San Martín de Porres
<http://www.usmp.edu.pe>. [En línea] 2002. [Citado el: 20 de 11 de 2010.]
<http://www.usmp.edu.pe/publicaciones/boletin/fia/info40/j2ee.html>.



- Organización, Visual Paradigm. 2009.** Sitio Web oficial Visual-Paradigm. [En línea] 2009. [Citado el: 19 de 11 de 2010.] <http://www.visual-paradigm.com/product/vpuml/>.
- Orinoco, Correo del. 2010.** Chávez: Homicidios han disminuido 52% en la parroquia Sucre. *Correo del Orinoco*. [En línea] 18 de 08 de 2010. [Citado el: 05 de 11 de 2010.] <http://www.correodelorinoco.gob.ve/judiciales-seguridad/chavez-homicidios-han-disminuido-52-parroquia-sucre/>.
- Oviedo, Universidad de.** Sitio Web de la E.U de Ingeniería Técnica Informática de Oviedo. [En línea] [Citado el: 14 de junio de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
- Piñeiro, Juan Medín y Figueras, Antonio García. 2006.** *Hacia una arquitectura con JavaServer Faces*. Sevilla : s.n., 2006.
- Policía, Consejo General de. Mayo, 2010.** *Colección BAQUIA: REGLAS MÍNIMAS DE ESTANDARIZACIÓN PARA LOS CUERPOS POLICIALES. 3. PASOS Y HUELLAS. Manual sobre Procedimientos Policiales*. Caracas, Venezuela : s.n., Mayo, 2010.
- . **Mayo, 2010.** *Colección BAQUIA: REGLAS MÍNIMAS DE ESTANDARIZACIÓN PARA LOS CUERPOS POLICIALES. 6. POR LAS CALLES DEL PUEBLO. Practiguía sobre el Sistema Integrado de Información y Dirección de las Operaciones Policiales*. Caracas, Venezuela : s.n., Mayo, 2010.
- Policía, Consejo General de la.** Consejo General de la Policía. [En línea] [Citado el: 26 de Octubre de 2010.] <http://www.consejopolicia.gob.ve/> .
- Policía, Consejo General de. 2010.** *Manual sobre Procedimientos Policiales*. Caracas, Venezuela : s.n., 2010.
- . **2010.** *Sesión 2: Policía Nacional Bolivariana y vigencia del Patrullaje Policial*. 2010.
- Pressman, Roger S. INGENIERÍA DEL SOFTWARE. Un enfoque práctico.** Sexta Edición.
- Puente, Wilson.** Técnicas de investigación. *Portal de Relaciones Públicas*. [En línea] [Citado el: 5 de 11 de 2010.] <http://www.rrppnet.com.ar/tecnicasdeinvestigacion.htm>.
- Ramírez, Jorge Amado Soria. 2011.** *Descripción de la Arquitectura de Software Proyecto CICPC-PNB*. 2011. SIIPOL 5.0.
- Ramón, Óscar Sánchez. 2006.** *Java Server Faces*. [En línea] 2006. [Citado el: 20 de 11 de 2010.] <http://ditec.um.es/ssdd/trabajos/0506/exposicionJSF.ppt>.



Ramos, Juan Alonso. Introducción a Ajax4jsf. . *Adictos al Trabajo*. [En línea] [Citado el: 4 de 12 de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.

Seguridad Pública. [En línea] [Citado el: 13 de junio de 2011.] <http://www.seguridadpublica.es/2008/10/el-modelo-espanol-funciones-y-competencias-de-las-fuerzas-y-cuerpos-de-seguridad-del-estado-de-las-policias-autonomicas-y-de-las-policias-locales-relaciones-entre-las-diferentes-fuerzas-y-cuerpos-d-2/>.

Soto, Carlos Bustos. Centro Nacional de Estudios de Orden Público. [En línea] [Citado el: 23 de 4 de 2011.] http://www.ceneop.cl/index.php?option=com_content&view=article&id=391:orden-publico&catid=98:news&Itemid=586.

Tibidabo Ediciones, S.A. Helios:Gestión Policial Local. [En línea] [Citado el: 21 de 11 de 2010.] <http://www.tibidaboediciones.com/helios/central.htm>.

Torrijos, Ricard Lou. Programación en Castellano. *Introducción a la Tecnología JavaServer Faces*. [En línea] [Citado el: 21 de 11 de 2010.] http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233.

—. **2003.** Programación en Castellano. *Introducción a la Tecnología JavaServer Faces* . [En línea] 2003. [Citado el: 22 de 11 de 2010.] http://www.programacion.com/articulo/introduccion_a_la_tecnologia_javascript_faces_233.

Velásquez, Hugo Acero. 2006. *Situación de violencia y delincuencia en Venezuela y concentración de delincuencia en Caracas*. Caracas, Venezuela : s.n., 2006.

Walls, Craig y Breidenbach, Ryan. 2005. *SPRING IN ACTION*. Greenwich : Manning Publications Co, 2005.





Anexo 1

CP Gestionar Planificación de Patrullaje

SC 1 Incluir Planificación de Patrullaje

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Incluir Planificación de Patrullaje.	Permite incluir una Planificación de Patrullaje en el sistema.	Muestra los datos de la Planificación de Patrullaje. Ver SC 2: Ver Datos de Planificación de Patrullaje.	Dentro de: Menú Principal/Vigilancia y Patrullaje/Registro/Planificación de Patrullaje 1 Seleccionar e introducir datos correspondientes a la nueva Planificación. 2 Presionar el botón “Adicionar” Se muestra una la tabla de planificaciones del mes con los siguientes datos de cada planificación: Día de la planificación. Turno de Patrullaje Grupo de Patrullaje Repetir los pasos 1 y 2 tantas veces como sea necesario. 3 Presionar botón “Incluir”.
EC 1.2: Cancelar inclusión de Planificación de Patrullaje.	Permite cancelar la inclusión de una Planificación de Patrullaje.	Elimina los datos introducidos o seleccionados. Regresa a la vista	Dentro de: Menú Principal/Vigilancia y Patrullaje/Registro/Planificación de Patrullaje



		anterior.	<p>1 Seleccionar e introducir datos correspondientes a la nueva Planificación.</p> <p>2 Presionar el botón “Adicionar”</p> <p>Se muestra una la tabla de planificaciones del rango con los siguientes datos de cada planificación:</p> <p>Día de la planificación</p> <p>Turno de Patrullaje</p> <p>Grupo de Patrullaje</p> <p>Repetir los pasos 1 y 2 tantas veces como sea necesario.</p> <p>3 Presionar botón “Cancelar”.</p>
EC 1.3: Existen datos incompletos.	Indica que existen datos incompletos.	<p>Muestra el mensaje de error: “Existen campos vacíos que son obligatorios, por favor, complete estos datos”.</p> <p>Resalta en color rojo los campos vacíos.</p>	<p>Dentro de: Menú Principal/Vigilancia y Patrullaje/Registro/Planificación de Patrullaje</p> <p>1 Seleccionar e introducir datos correspondientes a la nueva Planificación.</p> <p>2 Presionar el botón “Adicionar”</p> <p>Se muestra una la tabla de planificaciones del mes con los siguientes datos de cada planificación:</p> <p>Día de la planificación</p> <p>Turno de Patrullaje</p> <p>Grupo de Patrullaje</p>



			<p>Repetir los pasos 1 y 2 tantas veces como sea necesario.</p> <p>3 Presionar botón "Incluir".</p>
EC 1.4: Existen datos incorrectos.	Indica que existen datos incorrectos.	<p>Muestra el mensaje de error: "Existen campos escritos incorrectamente, por favor, rectifique estos datos".</p> <p>Resalta en color rojo los campos incorrectos.</p>	<p>Dentro de: Menú Principal/Vigilancia y Patrullaje/Registro/Planificación de Patrullaje</p> <p>1 Seleccionar e introducir datos correspondientes a la nueva Planificación.</p> <p>2 Presionar el botón "Adicionar"</p> <p>Se muestra una la tabla de planificaciones del mes con los siguientes datos de cada planificación:</p> <p>Día de la planificación</p> <p>Turno de Patrullaje</p> <p>Grupo de Patrullaje</p> <p>Repetir los pasos 1 y 2 tantas veces como sea necesario.</p> <p>3 Presionar botón "Incluir".</p>

Descripción de las variables:

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Fecha Inicio	Lista desplegable	No	Texto seleccionable
2	Fecha Fin	Lista desplegable	No	Texto seleccionable
3	Día de la	Lista desplegable	No	Texto seleccionable



	planificación			
4	Grupo	Lista desplegable	No	Texto seleccionable
5	Hora Inicio	Campo de texto	No	Cadena de caracteres alfanuméricos
6	Hora Fin	Campo de texto	No	Cadena de caracteres alfanuméricos