

“Universidad de las Ciencias Informáticas”

“Facultad 2”



**Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas**

Título:

“Modernización del Sistema del Centro de Información y Mando de
Patrulla. Subsistema de Administración y Configuración de Operaciones.”

Autores: Lisbet Fariña La O

David Hernández Hernández

Tutores: Ing. Wilson Alba Cal

Ing. Lermy García Mas

“Ciudad de La Habana, Cuba. Junio, 2010”

“Año del 52 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Lisbet Fariña La O

Autor

David Hernández Hernández

Autor

Ing. Wilson Alba Cal

Tutor

Ing. Lermey García Mas

Tutor

DEDICATORIA

A mis padres que han sido mi guía y fuente de inspiración.

A mi abuelito que estaría orgulloso de verme echa una ingeniera.

A mi novio Michel por su paciencia, apoyo y amor en todo este tiempo de lejanía.

Lisbet.

A mis padres, por ser mi fuente de inspiración y mi mayor orgullo.

A mi hermano Daniel.

A toda mi familia, por apoyarme siempre.

A Lily, mi novia querida... por ser mi punto de apoyo más firme...

David

AGRADECIMIENTOS

A mis padres que han sido mi guía, mi mano amiga, mi apoyo y mi fuente de inspiración. Todo lo que soy hoy se los debo a ellos, los quiero mucho.

A mi novio Michel por todo su amor y apoyo en todo este tiempo.

A mis hermanos, mi sobrinita y mi prima Yuselis por estar ahí siempre para mí.

A mis primos Yuri y Papito por ayudarme en los momentos más difíciles.

A el equipo completo de patrulla y en especial a todas las niñas que sin su ayuda el desarrollo de este trabajo no hubiese sido posible.

A los profesores : Yunisel, Rosa, Susana, Raciél y Pla que siempre estuvieron guiándonos en todo este tiempo de trabajo.

A mis tutores Lermey y Wilson que estuvieron ahí siempre que los necesitamos.

A David por su apoyo y paciencia.

A todos los que conformaron el tribunal por su ayuda y sugerencias.

A todas mis amistades en especial: May, Elena, Yaima, Yanira y Franyudey por ofrecerme su cariño y apoyo en todo este tiempo.

A todos los profesores que permitieron que en estos años me formara profesionalmente.

A la UCI por acogerme y ser parte de esta gran familia.

A la Revolución y Fidel por darme la oportunidad de cumplir mis sueños.

A todos muchas gracias!!!

Lisbet

AGRADECIMIENTOS

Mi mayor agradecimiento es para mis padres: mi mamá, la persona más dulce y noble que conozco, por todo su amor, su dedicación y por todos los esfuerzos y sacrificios que ha realizado, que me han permitido llegar hasta aquí, muchas gracias. A mi papá, que aún cuando la vida no lo dio la oportunidad de estar presente físicamente para verme con el Título de Ingeniero, ha sido siempre mi motor impulsor, mis ganas de superarme, mi empeño para ser una mejor persona cada día. A mi hermano Daniel, por ser mi guía, mi apoyo, mi segundo padre, por los momentos tan duros que me ayudó a superar, muchas gracias. A toda mi familia: mis abuelos, a mima Tomasa, mis tíos, mis primos, mi sobrinita Elis.

A mi novia Lily, por soportar heroicamente a mi lado. Muchas gracias por todo el apoyo que me brindó, y por todo su amor,

A mis amigos Yusley, Dairon, Yuri y todos aquellos que me han soportado durante tanto tiempo, a mis tutores, a mi compañera de tesis, al equipo de desarrollo del proyecto, a la Revolución. En fin a todos los que de una forma u otra tuvieron que ver con la creación de la tesis, Muchas Gracias

David

RESUMEN

Dentro de los principios fundamentales en la constitución de toda nación y principalmente en Cuba figura el de la seguridad ciudadana, la protección de la población, el logro del bienestar público y el predominio del orden. Es responsabilidad de los Órganos de Seguridad del estado que estos principios se cumplan y que la población quede satisfecha con el servicio prestado.

El Centro de Información y Mando de la Unidad Provincial de Patrullas con el fin de fortalecer su efectividad en la dirección del enfrentamiento de los distintos hechos delictivos que afectan la tranquilidad y el orden social, suscita la creación del Sistema del Centro de Información y Mando como un sistema automatizado encargado de la gestión del Puesto de Mando y el Centro de Información de dicha unidad. El mismo constituirá un perfeccionamiento integral del sistema informático existente y estará compuesto por varios subsistemas, entre ellos el de Administración y Configuración de Operaciones, cuya construcción será el objetivo general del presente trabajo.

El Subsistema de Administración y Configuración de Operaciones es el encargado de gestionar las configuraciones generales del sistema, para ello se encarga de gestionar la información de las entidades que se manipulan, los nomencladores utilizados y las configuraciones de seguridad del sistema.

Este trabajo ofrece una explicación sobre las distintas herramientas utilizadas y los elementos necesarios para la construcción de la aplicación de Administración y Configuración de Operaciones, que posee un conjunto de funcionalidades que le dan solución a las necesidades del centro.

Palabras Clave: Administración, Configuración, Subsistema, Centros de emergencias

TABLA DE CONTENIDOS

<i>INTRODUCCIÓN</i>	1
<i>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</i>	5
1.1 Introducción	5
1.2 Administración	5
1.3 Configuración Informática	5
1.4 Centros de Gestión de Emergencias	6
1.5 Números Telefónicos de Emergencia	6
1.6 Sistemas de Gestión de Emergencias Existentes	7
1.6.1 CeCoCo – Centro de Control de Teltronic.....	7
1.6.2 CeCo-SEC- Para Seguridad Pública, Emergencias y Bomberos.....	7
1.6.3 Sistema Informático para la Gestión de Emergencias (SIGE)	7
1.6.4 Sistema de Gestión de Emergencia (SIGEME)	8
1.6.5 Sistema de Gestión de Emergencias de Seguridad Ciudadana, SIGESC.	8
1.6.6 Sistema del Número Único de Seguridad y Emergencia 123	9
1.6.7 Sistema Automatizado del Puesto de Mando de la Unidad Provincial de Patrulla	9
1.7 Metodología de Desarrollo de Software	10
1.7.1 Proceso Unificado de Desarrollo de Software (RUP)	11
1.8 Herramientas CASE.....	13
1.8.1 Visual Paradigm para UML	14
1.9 Plataforma de Desarrollo de Software.....	14
1.9.1 Plataforma. Net	14
1.9.2 Entity Framework	17
1.9.3 IDE: Microsoft Visual Studio 2008.....	18
1.9.4 Lenguaje de programación C# 3.0	19
1.10 CAGWPFS.....	19
1.11 Sistema Gestor de Base de Datos Oracle 11g	20
1.12 Patrones.....	21
1.13 Arquitectura de Software	22
1.13.1 Arquitectura del Sistema del Centro de Información y Mando.	22

1.13.2 Arquitectura de seguridad del Sistema del Centro de Información y Mando.....	23
1.14 Conclusiones.....	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SUBSISTEMA	26
2.1 Introducción	26
2.2 Modelo de Dominio	26
2.3 Requerimientos Funcionales.....	28
2.4 Requisitos no funcionales	34
2.5 Descripción de la Solución Propuesta.....	35
2.6 Modelos de Caso de Uso del Subsistema.....	36
2.6.1 Diagrama de Paquetes	37
2.6.2 Diagrama de Caso de Uso	37
2.7 Caso de Uso Expandido	41
2.8 Conclusiones	45
CAPÍTULO 3: DISEÑO DEL SUBSISTEMA.....	46
3.1 Introducción	46
3.2 Arquitectura del Subsistema.	46
3.3 Diagrama de Paquetes del Diseño.....	48
3.4 Diagrama de Clases de Diseño.....	49
3.5 Diagrama de Clases Persistentes	52
3.6 Modelo de Datos.....	53
3.7 Conclusiones	54
CAPÍTULO 4: IMPLEMENTACIÓN DEL SUBSISTEMA.....	55
4.1 Introducción	55
4.2 Modelo de Implementación	55
4.2.1 Diagrama de Componentes	55
4.3 Descripción de Componentes	56
4.4 Diagrama de Despliegue	58
4.5 Conclusiones	58
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD	59

5.1 Introducción	59
5.2 Estimación por Punto de Caso de Uso.....	59
5.2.1 Cálculo de Punto de Caso de Uso sin Ajustar	59
5.2.2 Cálculo de Punto de Caso de Uso Ajustados	61
5.3 Cálculo del Esfuerzo en Horas-Hombre	63
5.3.1 Distribución del Esfuerzo entre las diferentes actividades del Subsistema.....	64
5.3.2 Cálculo del costo del Sistema del Centro de Información y Mando	64
5.3.3 Cálculo del tiempo de desarrollo del Subsistema de Administración y Configuración de Operaciones	65
5.4 Beneficios tangibles e intangibles	65
5.5 Análisis de costo.....	66
5.6 Conclusiones	66
<i>CONCLUSIONES GENERALES</i>	67
<i>RECOMENDACIONES</i>	68
<i>REFERENCIA BIBLIOGRÁFICA</i>	69
<i>BIBLIOGRAFÍA</i>	71

INTRODUCCIÓN

La seguridad ciudadana es una condición necesaria para el funcionamiento de la sociedad y uno de los principales criterios para asegurar la calidad de vida, por tal motivo cada país dentro de su constitución referencia este tema como un punto de vital importancia. En todo el mundo existen un sin número de centros y entidades (estaciones de policías, unidades de bomberos y la cruz roja) que tienen la misión de actuar en defensa de los ciudadanos, respondiendo a sus necesidades e intereses, y brindar un servicio público capaz de garantizar condiciones estables y favorables en la dinámica social.

El Centro de Información y Mando de la Unidad Provincial de Patrullas (en lo adelante CIMUP) es una de las instituciones presentes en Cuba que contribuye a que la población sienta que está protegida ante los problemas que atentan contra su seguridad. Este es un centro de referencia que hace uso de tecnología de punta con el fin de garantizar la dirección operativa de las fuerzas y medios motorizados de la Unidad Provincial de Patrulla (en lo adelante UPP), Puntos de Control y Tránsito. Dentro de las funciones operativas que realiza se encuentran:

- Dirigir el servicio motorizado de vigilancia y patrullaje en la ciudad, puntos de control y carreteras aledañas.
- Brindar asistencia y respuesta a las llamadas de emergencias de la población (106).
- Controlar el cumplimiento de los planes y servicios operativos diseñados ante eventos con participación masiva de la población.
- Monitorear el funcionamiento de la red semafórica de la ciudad.

Esta institución cuenta con varias áreas de trabajo que se apoyan de un sistema informático encargado de garantizar el cumplimiento de las misiones asignadas. Uno de estos sistemas es el del Centro de Información y Mando que integra varios procesos importantes que ayudan a dar seguimiento a la misión fundamental del CIMUP: la prevención y el enfrentamiento a la delincuencia, la indisciplina vial y otros sucesos que afectan la tranquilidad y el orden social. Dentro de los procesos que integra se encuentran la atención a las llamadas de emergencias, la dirección operativa de las patrullas que se mantienen en la vía, mantener información visual y en tiempo real de los carros que se encuentran patrullando; y la ubicación geográfica de los hechos delictivos, planes especiales y objetivos de interés policial. El centro cuenta además con el Sistema Integral de Trabajo de la Unidad de Patrullas (SITUP) encargado de controlar la disponibilidad de los recursos humanos y materiales, la información general sobre el desempeño de las fuerzas y el estado de la unidad para el cumplimiento de las misiones confiadas. Otro

de los sistemas es el Informativo, encomendado a realizar recuperaciones temáticas, análisis de información, generación de decisiones y la diseminación de información operativa, inmediata y uniforme a todo el medio policial.

En estudios realizados se detectó que a pesar de las mejoras que ha recibido en varias oportunidades el Sistema del Centro de Información y Mando, este continúa presentando problemas. Esto se debe a la imposibilidad de acceso al sistema desde unidades externas de interés (Puntos de Control y Tránsito), lo que dificulta su sostenimiento. No define normas o parámetros que guíen el buen funcionamiento de la institución provocando que las respuestas a las situaciones de emergencias que se presenten no sean las más rápidas y confiables. Además, el sistema no permite elaborar las variantes de distribución de las áreas de patrullaje trayendo consigo conflictos a la hora de realizar las asignaciones de los carros patrulleros a estas áreas. Los nomencladores existentes en la aplicación no están organizados por categorías, como consecuencia los operadores del sistema pierden tiempo al realizar las acciones elementales en el mismo. A causa de las continuas modificaciones que ha recibido el software no se tiene de manera organizada la información referente a los puestos de trabajo y los usuarios que acceden al sistema, por tal motivo el control de las acciones que realizan estos últimos en sus puestos de trabajo no se realiza eficientemente.

La situación antes descrita conduce al siguiente **problema científico**: ¿Cómo gestionar los parámetros de configuración y funcionamiento del Sistema del Centro de Información y Mando de Patrulla?

El **objeto de estudio** de esta investigación lo constituyen los procesos de configuración y funcionamiento para los sistemas automatizados de un centro de emergencia. Partiendo de lo anterior se reconoce como **campo de acción** los procesos de configuración y funcionamiento del Sistema del Centro de Información y Mando de Patrulla.

Para darle solución al problema planteado se definió como **objetivo general**: El análisis, diseño e implementación de un Subsistema de Administración y Configuración de Operaciones como parte del Sistema del Centro de Información y Mando de Patrulla. Consecuentemente se definieron los siguientes **objetivos específicos**:

- Identificar los parámetros de configuración general que rigen el funcionamiento del Subsistema de Administración y Configuración de Operaciones.
- Recopilar la información de las entidades relacionadas con la seguridad y el funcionamiento general del sistema.

- Identificar y clasificar los nomencladores que se utilizarán en el Subsistema de Administración y Configuración de Operaciones.
- Implementar el software correspondiente al Subsistema de Administración y Configuración de Operaciones.

La **idea a defender** es: Con el desarrollo del Subsistema de Administración y Configuración de Operaciones se pretende cumplir con las exigencias del cliente y dar solución a los problemas de configuración y funcionamiento del Sistema del Centro de Información y Mando de Patrulla.

De acuerdo con el objetivo general se trazaron las siguientes **tareas de la investigación**:

- Caracterización de las herramientas, los lenguajes de programación, la metodología de desarrollo de software y la plataforma de desarrollo a utilizar durante la construcción del software.
- Identificación de mecanismos para la administración y configuración de operaciones.
- Identificación de los principales conceptos que intervienen en los procesos de negocio del Subsistema de Administración y Configuración de Operaciones y las relaciones entre ellos.
- Elaboración e identificación de los requerimientos tanto funcionales como no funcionales.
- Modelación de los casos de uso que describen las funcionalidades que debe tener el software.
- Definición de un diseño acorde a los requisitos funcionales y no funcionales.

Métodos Científico de Investigación

Los métodos científicos usados como apoyo a la investigación de este trabajo fueron los siguientes:

Métodos Teóricos:

- **Análítico-Sintético:** con el fin de definir y valorar la metodología de desarrollo de software, las herramientas y las tecnologías necesarias para el desarrollo del subsistema.

Métodos empíricos:

- **Observación:** utilizado con la finalidad de caracterizar y analizar cómo se realizan actualmente los diferentes procesos de negocio en el Sistema del Centro de Información y Mando.
- **Entrevista:** utilizado para realizar entrevistas tanto individuales como colectivas a los distintos trabajadores del Puesto de Información y Mando de la Unidad de Patrullas para identificar los procesos que se llevan a cabo en este centro.

Este trabajo se encuentra distribuido en 4 partes esenciales: Resumen, Introducción, Desarrollo y Conclusiones. El Desarrollo está dividido en 5 capítulos:

Capítulo 1. Fundamentación Teórica: Se describen detalladamente los distintos conceptos que se manejan para el desarrollo del Subsistema de Administración y Configuración de Operaciones. Se realiza

una investigación general de los distintos sistemas de gestión de emergencia existentes; además de un estudio de las diferentes herramientas, metodologías de desarrollo de software, lenguajes de programación y plataformas a utilizar en la construcción de la aplicación.

Capítulo 2. Características del Subsistema: Se describen detalladamente los distintos conceptos de negocio referentes a la administración y configuración de operaciones, a partir de estos se especifican los requisitos funcionales y no funcionales que contendrá el subsistema. Partiendo de lo anterior se modelan y describen los casos de uso del subsistema, obteniéndose al final un diseño de los prototipos de interfaces iniciales.

Capítulo 3. Diseño del Subsistema: Se transforman los requerimientos funcionales y no funcionales en modelos de clases del diseño que serán la base para la implementación.

Capítulo 4. Implementación del Subsistema: Describe la transformación de los elementos del diseño en componentes, ficheros de código fuente, ejecutables y las dependencias entre ellos.

Capítulo 5. Estudio de Factibilidad: Se realiza un estudio de factibilidad a partir de la estimación del tamaño y del esfuerzo necesario para realizar el Subsistema de Administración y Configuración de Operaciones a través del método de estimación Puntos de Caso de Uso.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el desarrollo de este capítulo se analizan los principales aspectos y conceptos teóricos que guardan relación con la investigación cuya comprensión es necesaria para un mejor entendimiento del problema a resolver. Es por ello que se abordarán temas relacionados con la administración, la arquitectura de software, la configuración de operaciones y centros de emergencias. Se describen además las características principales de la metodología de desarrollo de software, lenguaje de programación, tecnologías y herramientas que se van a tener en cuenta para el desarrollo de la aplicación, justificando finalmente la selección realizada para el Subsistema de Administración y Configuración de Operaciones a desarrollar.

1.2 Administración

Según la definición de varios autores la administración es la actividad de planeación, organización, ejecución y control, desempeñada para determinar y alcanzar los objetivos con el uso de seres humanos y otros recursos (materiales y económicos). (1)

En el proceso de construcción de un software el concepto de administración es un elemento trascendental para su adecuado funcionamiento, pues facilita que el sistema funcione de una manera ordenada, segura y consistente.

Dentro de un sistema informático la administración se compone por un módulo o subsistema que proporciona diferentes funcionalidades administrativas que permiten:

- Controlar las operaciones que garanticen la seguridad.
- Definir las estructuras lógicas con el fin de tener una buena organización del sistema.
- Gestionar las configuraciones globales o específicas de los negocios que integra.
- Gestión de usuarios y estaciones de trabajo.

1.3 Configuración Informática

La configuración informática es el grupo de datos e información que caracterizan diferentes elementos de una computadora, como son programas, aplicaciones o elementos de hardware / software. Se conforma por el conjunto de aparatos y programas que constituyen un sistema informático.

Partiendo del concepto anterior es preciso considerar entonces que la configuración informática es la manera de adaptar una aplicación de software al resto de los componentes del entorno de desarrollo y a

las necesidades del usuario. Por tal motivo, es una tarea esencial antes de trabajar con un nuevo sistema automatizado.

La tendencia que existe actualmente para reducir las necesidades de configuración es hacer uso de sistemas que permitan detectar en que entorno se va a instalar el nuevo elemento, configurándose automáticamente sin la necesidad de requerir la participación del usuario y en caso de ser necesaria se intenta facilitar al máximo el proceso de configuración.

1.4 Centros de Gestión de Emergencias

Los Centros de Gestión de Emergencias son aquellas instituciones responsables de atender las situaciones de emergencias que ocurran en una zona o localidad determinada. Su objetivo fundamental es proveer a la población de un alto grado de seguridad, garantizando la atención inmediata.

Muchos de estos centros utilizan Sistemas de Gestión de Emergencias para un mejor funcionamiento. Estos sistemas automatizados de gestión son los encargados de la recepción y despacho de las solicitudes de emergencia. Algunos poseen tecnologías avanzadas para aumentar la rapidez y confiabilidad de las respuestas ante un suceso (telefonía, radio, sistemas de cómputo, GPS, mapificación en tiempo real y grabación de llamadas).

1.5 Números Telefónicos de Emergencia

En todo el mundo existen números telefónicos que pueden ser utilizados para solicitar ayuda ante cualquier caso de emergencia. Algunos de ellos son mundialmente conocidos como es el caso del 911 en los Estados Unidos, y que extiende su uso a países como Costa Rica, Paraguay, Canadá, Argentina, Uruguay, El Salvador, Panamá y Ecuador. El 112 es el principal número de emergencia utilizado en la Unión Europea específicamente en el centro de emergencia de Madrid (centro de referencia en toda Europa). Aunque cabe destacar que son usados otros números como el 911 y 999, este último es utilizado también en los países de África y Asia.

En el continente latinoamericano existen otros exponentes de estos números telefónicos, aparte del 911, como es el caso del 123, número único de Seguridad y Emergencias de la Ciudad de Medellín, Colombia y el 171 en Venezuela. Actualmente en Cuba, en Ciudad de la Habana, existen 2 números telefónicos para la gestión de emergencias el 106 (el más usado) donde los ciudadanos pueden informar cualquier incidente a la Policía Nacional Revolucionaria y el 105 que es el número de los Bomberos.

1.6 Sistemas de Gestión de Emergencias Existentes

Gran parte de los centros de emergencias existentes en todo el mundo se apoyan en Sistemas de Gestión de Emergencias con el fin de ser más efectivos a la hora de dar respuesta a un incidente determinado. Algunos de estos sistemas se describen a continuación:

1.6.1 CeCoCo – Centro de Control de Teltronic

Los sistemas CeCoCo se destacan por su rendimiento y fiabilidad, permiten la realización de distintas configuraciones que se adaptan a distintos ámbitos profesionales tales como (2):

- Centros de Control para Seguridad Pública, Emergencias y Bomberos.
- Centros de llamada 911 y 112.
- Centros de Gestión de Transporte Público.
- Despachadores vía línea de propósito general.

1.6.2 CeCo-SEC- Para Seguridad Pública, Emergencias y Bomberos

Este sistema se enfoca a la gestión de recursos, el mantenimiento de llamadas y la adquisición de datos de incidentes; siendo una de sus principales ventajas la integración automatizada de todas estas funciones.

Entre los rasgos destacables de la solución se encuentran:

- Las interfaces multivoz y de comunicación mediante datos (TETRA, audio analógica, telefonía, radiobúsqueda y GSM (Sistema Global de Comunicaciones Móviles)).
- Registro de llamadas y acceso rápido para reproducir la última llamada.
- Gestión de guías telefónicas.
- Muestra alarmas configurables por el usuario.
- Indicación de la asignación de recursos a los incidentes.
- Acceso completo a los registros del incidente.
- Administración web.

1.6.3 Sistema Informático para la Gestión de Emergencias (SIGE)

El SIGE se conforma por una serie de sistemas que tratan tanto la gestión de los diferentes tipos de información, como la integración de esos sistemas en un entorno común. Además, permite el desarrollo de las comunicaciones entre los diferentes órganos implicados en el proyecto.

Entre las funcionalidades que brinda este sistema se encuentran:

- Emisión de alertas en tiempo real de acuerdo con la información recogida por las redes de detección.
- Elaboración de informes y estadísticas gráficas y alfanuméricas.
- Seguimiento de actuaciones.
- Gestión de la información sobre medios y recursos disponibles.
- La interfaz de usuario de este sistema está desarrollada en Visual Basic con el objetivo de cumplir con los requisitos de representación cartográfica.

1.6.4 Sistema de Gestión de Emergencia (SIGEME)

Es una plataforma modular multicanal de alta disponibilidad para la gestión de actuaciones, comunicaciones y recursos móviles en caso de emergencia, equipada con las más avanzadas tecnologías y prestaciones (3). Facilita la comunicación entre interlocutores que usan sistemas de comunicación distintos e independientes entre sí. Visualiza y localiza geográficamente los lugares donde ocurren los diferentes sucesos, así como los distintos puntos de interés (Ejemplo: hospitales y escuelas) por medio de mapas y callejones electrónicos.

Es una herramienta concebida para la administración pública y orientada a coordinar a los centros de emergencias que utilizan el 112, brigadas forestales y bomberos.

1.6.5 Sistema de Gestión de Emergencias de Seguridad Ciudadana, SIGESC.

El SIGESC es un sistema informático que proporciona aplicaciones con una alta integración entre ellas, las cuales permiten gestionar de forma eficiente las actividades relacionadas con la atención de las solicitudes de emergencias formuladas por la población venezolana.

Se encuentra dividido en las siguientes aplicaciones:

- Recepción de Llamadas.
- Despacho.
- Supervisión.
 - Supervisión de Operadores.
 - Supervisión de Despacho.
 - Supervisión General.
- Administración y Control de Recursos.
- AVL y Sincronización.
- Estadísticas.

- Administración Informática y Configuración.

1.6.6 Sistema del Número Único de Seguridad y Emergencia 123

Es un sistema informático creado por la Unión Temporal Colombo-Canadiense Procálculo y es utilizado por el Centro de Emergencia de Medellín, Colombia. Este sistema está compuesto por diferentes componentes que se mencionan a continuación (4):

- **Planta telefónica:** encargada de recibir las llamadas que entran por 123 y despacharlas automáticamente a cada uno de los receptores.
- **PowerCAD:** es una aplicación sistematizada para agencias de respuestas de emergencias que facilita el manejo de los incidentes y los recursos.
- **Power123:** es una aplicación de software en el Sistema Operativo Windows NT que se encarga de recepcionar y clasificar las llamadas de emergencias.
- **PowerMap:** es una aplicación que permite la ubicación geográfica de la información que brindan las personas que realizan las llamadas y de los incidentes activos, además de los niveles geográficos necesarios para la atención de la emergencia.

1.6.7 Sistema Automatizado del Puesto de Mando de la Unidad Provincial de Patrulla

Este sistema fue creado en Ciudad de La Habana en el año 2000, con el objetivo de elevar la efectividad de la vigilancia y patrullaje de la capital y potenciar las capacidades del Puesto de Mando y las unidades subordinadas.

Dentro de sus características principales se encuentran (5):

- El uso de la Red Interna y la Red del MININT para el intercambio informativo con las Regiones y Estaciones de Seguridad Pública.
- Actualización en tiempo real de los eventos necesarios para el funcionamiento del Puesto de Mando.
- Empleo de mapas, localización de móviles con GPS (Sistema de Posicionamiento Global) e imágenes, con el objetivo de apoyar la toma de decisiones ante las diferentes situaciones del trabajo diario y contingencias.
- Recuperaciones temáticas sobre la información almacenada para los usuarios internos del Puesto de Mando y mediante tecnología Web disponibles en el Sitio Informativo del Puesto de Mando.

Los sistemas de emergencias internacionales descritos con anterioridad fueron creados a partir de los requerimientos de distintas instituciones que se encuentran enfocadas a la seguridad ciudadana. Por

tanto, implantar una de estas aplicaciones como propuesta de solución no es lo más conveniente debido a que los requerimientos que se definen en estos sistemas no son compatibles con los requeridos por el cliente del sistema a desarrollar.

El sistema de gestión de emergencia que actualmente se encuentra en explotación en Cuba presenta varias deficiencias. Esto se debe a que muchas de las funcionalidades que el mismo presenta se encuentran desactivadas y desactualizadas, además, el sistema no presenta parámetros de configuración que lo guíen a un buen funcionamiento y aún con las mejoras que ha recibido, las aplicaciones que lo conforman no se encuentran integradas en un todo, provocando que las acciones fundamentales que se llevan a cabo por parte de los operadores no sean las más efectivas.

1.7 Metodología de Desarrollo de Software

El éxito de todo desarrollo de software depende en gran medida de la metodología de desarrollo escogida. Esto se debe a que las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Las metodologías de desarrollo de software se encargan de elaborar estrategias de desarrollo que promueven prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (6). Indica paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Detalla la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Al elegir la metodología a seguir para el proceso de desarrollo que se va a emprender se deben tener en cuenta aspectos como: la dimensión o el tamaño del proyecto, el tiempo de duración y la complejidad. Partiendo de esto las metodologías se pueden clasificar en estructuradas, orientadas a objetos y en sistemas de tiempo real. Otros de los términos por el cual se conocen actualmente es por metodologías tradicionales y ágiles.

Las metodologías tradicionales o robustas están diseñadas para proyectos complejos, con un gran número de personas involucradas. Se centran en el control del proceso, estableciendo las actividades

involucradas, los artefactos a producir y las herramientas y notaciones que se utilizarán. Entre estas se encuentran RUP¹ y MSF².

Las metodologías ágiles brindan gran peso a la interacción con el cliente y al desarrollo incremental y acelerado del software en iteraciones muy cortas, contando generalmente con un equipo pequeño de desarrolladores. Tienen como base de sus resultados la comunicación con los usuarios involucrados en el proceso. Entre sus exponentes se encuentran XP³, AUP⁴ y RAP⁵.

1.7.1 Proceso Unificado de Desarrollo de Software (RUP)

El Rational Unified Process o el Proceso Unificado de Desarrollo de Software es una metodología tradicional cuyo surgimiento es el resultado de varios años de desarrollo y uso práctico, en el que se han unificado técnicas de desarrollo, a través del UML (Lenguaje Unificado de Modelado) (7). Es la metodología estándar más utilizada para el análisis, implementación y documentación de los sistemas orientados a objetos.

RUP define 9 flujos de trabajo principales. Los 6 primeros conocidos como flujos de ingeniería y los 3 últimos de apoyo. (7)

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba:** Busca los defectos a lo largo del ciclo de vida.

¹ RUP: Proceso Unificado de Desarrollo de Software.

² MSF: Microsoft Solutions Framework.

³ XP: Programación extrema.

⁴ AUP: Proceso Ágil Unificado.

⁵ RAP: Desarrollo Rápido de Aplicaciones.

- **Despliegue:** Produce una liberación del producto y realiza actividades (empaquete, instalación y asistencia a usuarios) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización / actualización concurrente de elementos, control de versiones, evolución del software y control de configuración.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto.

El proceso de desarrollo en RUP se divide en ciclos, obteniendo un producto al final de cada uno de ellos, estos a la vez se dividen en 4 fases (8):

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen.
- **Construcción:** Obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida de RUP se caracteriza por ser:

- **Dirigido por caso de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios para comprenderlo, desarrollarlo y producirlo económicamente. RUP se

desarrolla mediante iteraciones, comenzando por los casos de usos relevantes desde el punto de vista de la arquitectura.

- **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y el diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Aunque cada iteración tiene que proponerse un incremento en el proceso de desarrollo, todas deben aportar al principal resultado de la fase en la que se desarrolla.

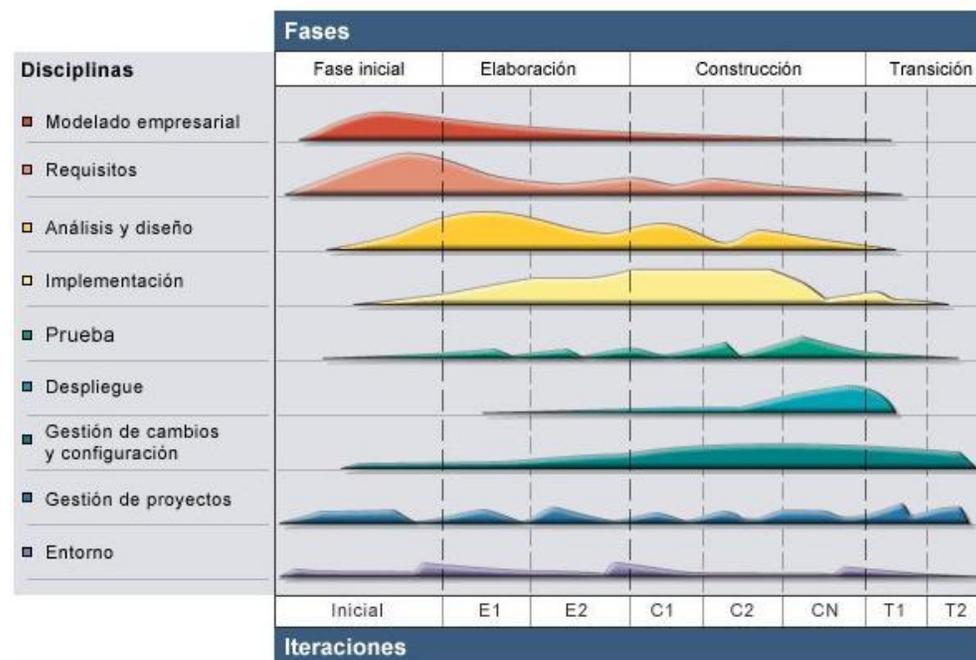


Fig. 2 Ciclo de vida de RUP.

1.8 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, en español Ingeniería de Software Asistida por Computadoras) son aplicaciones informáticas que tienen como finalidad el aumento de la productividad en el desarrollo del software reduciendo el costo de la mismas en tiempo y dinero (9). Dentro de sus objetivos se encuentran aumentar la calidad del software, el mantenimiento de los sistemas informáticos, mejorar la planificación del proyecto, la reutilización del software y estandarización de la

documentación, generación de código, pruebas de errores y gestión de proyectos. Del listado de las aplicaciones CASE más usadas se encuentra el Visual Paradigm para UML.

1.8.1 Visual Paradigm para UML

Visual Paradigm es una herramienta CASE profesional que utiliza UML 2.1 como lenguaje de modelado visual y soporta el ciclo de vida completo de desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Dentro de sus características fundamentales se encuentran:

- Genera documentación y código a partir de los diagramas de clases.
- Se integra con diferentes entornos de desarrollo como el Visual Studio, Eclipse, NetBeans y Borland JBuilder.
- Facilita una rápida construcción de aplicaciones de mayor calidad con un menor coste permitiendo dibujar todos los tipos de diagramas.
- Realiza el modelado en colaboración y simultáneamente con el Visual Paradigm TeamWork Server y Subversion.

1.9 Plataforma de Desarrollo de Software.

Se entiende como Plataforma de Desarrollo de Software el principio de hardware o software sobre el cual una aplicación es compatible y puede ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución.

1.9.1 Plataforma. Net

La plataforma .NET permite el desarrollo de aplicaciones completamente orientadas a objetos, seguras y sencillas de instalar. Es un conjunto de tecnologías de software que facilitan el desarrollo de aplicaciones de escritorio (Windows), web (ASP.Net), bibliotecas de código reutilizables (dlls), controles de usuarios y acceso a datos que se ejecutan bajo el Framework .Net.

1.9.1.1 Framework .Net

El Framework .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias es posible desarrollar cualquier tipo de aplicación con cualquiera de ellos. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C# (C Sharp), Visual Basic y C++ hasta otros menos conocidos como Perl y Cobol.

Mediante este marco de trabajo se ofrece un entorno de ejecución altamente distribuido que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- Bibliotecas (librerías) de clases de .Net.
- CLR (Common Language Runtime, en español Entorno Común de Ejecución para Lenguajes).



Fig. 3 Componentes de Framework .Net.

El CLR es el núcleo del Framework .Net ya que es el entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes de programación, ampliando el conjunto de servicios que ofrece el sistema operativo estándar Win32. Entre los elementos que administra se encuentran: la memoria, la ejecución de subprocesos, la ejecución del código, la comprobación de la seguridad del código y la compilación. Estas características son específicas del código administrado que se ejecuta en el mismo.

Las bibliotecas de clases del Framework .Net son una colección de tipos (clases, estructuras, enumeradores, delegados e interfaces) reutilizables que se integran estrechamente con el CLR. La manera de organizarla dentro del código es por medio de los espacios de nombre, donde cada clase esta organizada por espacios de nombre según su funcionalidad. Incluye tres componentes fundamentales:

- ASP. NET⁶ para el desarrollo de aplicaciones y servicios web.
- Windows Forms para la construcción de interfaces de usuario.
- ADO. NET para conectar las aplicaciones a las bases de datos.

⁶ ASP. NET: es un framework para el desarrollo de aplicaciones web, sitios web dinámicos y servicios web XML.

1.9.1.2 Framework .Net 3.5

El Framework .Net en su versión 3.5 se basa en las versiones 2.0, 3.0 y sus Service Packs correspondientes. Incluye nuevos ensamblados que proporcionan nuevas funcionalidades a las anteriores versiones.

Dentro de las características nuevas y mejoradas con que cuenta Framework .Net 3.5 se encuentran:

- **Windows Presentation Foundation** (WPF, siglas en inglés), es la última tecnología de Microsoft para la presentación de aplicaciones tanto de escritorio como web. Esta tecnología establece una clara separación entre el diseño de una interfaz y su comportamiento, de forma tal que puedan trabajar de manera independiente diseñadores y programadores para obtener un mejor resultado final. Permite obtener aplicaciones mucho más atractivas visualmente, además de la inclusión de conceptos como los commands y el binding de datos de forma simple y declarativa, los cuales permiten un desarrollo mucho más ágil. Combina las interfaces de aplicación, gráficos 2D y 3D, documentos y multimedia en el mismo framework. Su motor de procesamiento basado en vectores utiliza la aceleración del hardware de tarjetas gráficas modernas. Esto hace que las interfaces de usuario sean más rápida, escalable y de resolución independiente.
- **Windows Communication Foundation** (WCF, siglas en inglés), es la unión de todas las tecnologías de Microsoft para la comunicación entre aplicaciones bajo un único estilo de desarrollo. Antes de la aparición de WCF las aplicaciones tenían que ser conscientes de los mecanismos que utilizaban para la comunicación entre ellas, de forma tal que si se deseaban cambiar debían refactorizar prácticamente toda la aplicación. Con WCF basta con establecer un contrato de servicio y luego definir (a nivel de configuraciones) cual es el mecanismo que se desea utilizar para la comunicación, de esta forma en caso de ser necesario un cambio solo se cambia la configuración, pues la aplicación en ningún momento es consciente del mecanismo que se utiliza. Unifica una gran variedad de funcionalidades de sistemas distribuidos en una arquitectura organizable y extensible, que abarca transportes, sistemas de seguridad, patrones de mensajería, sistemas de codificación, topologías de red y modelos de alojamiento.
- **Language Integrated Query** (LINQ, siglas en inglés), es la última tecnología de Microsoft para la realización de consultas estilo SQL (Lenguaje de Consulta Estructurada) a colecciones de objetos. LINQ permite un desarrollo mucho más ágil, puesto que no es necesario realizar algoritmos complejos y propensos a errores para el tratamiento de colecciones de objetos, sino que basta con realizar una consulta al estilo SQL sobre la colección para obtener el resultado deseado. Todo esto apoyado del chequeo de tipos y el IntelliSense del Visual Studio.

1.9.2 Entity Framework

El Entity Framework es la evolución del ADO.NET⁷ para realizar mapeos objeto-relacional. No solo es un modelo objeto-relacional sino que provee herramientas mucho más potentes como es el entity composition, el cual permite crear una entidad lógica (capa a nivel de objetos) partiendo de varias entidades físicas (tablas de la Base de Datos). Permite a los programadores trabajar con datos en forma de objetos y propiedades específicos del dominio sin tener que pensar en las tablas de las bases de datos y en las columnas en las que se almacenan los mismos. Para esto, se eleva el nivel de abstracción en la que los programadores pueden trabajar al tratar con datos y se reduce el código requerido para crear y mantener las aplicaciones orientadas a datos.

La pieza central del Entity Framework es el Entity Data Model (EDM), el mismo describe el esquema de diseño, que se usa para generar las clases programables que usa el código de la aplicación. Cuenta además con el componente Objeto de Servicio, el cual facilita a los desarrolladores trabajar con las clases del CLR generadas a partir del modelo conceptual. El LINQ to Entities es otra de las facilidades que brinda el Entity Framework pues permite a los programadores escribir consultas en la base de datos utilizando uno de los lenguajes de programación del Framework .Net admitidos, como Visual Basic o Visual C#. Por último, se debe mencionar que el mismo incluye un conjunto de herramientas en evolución que generan asignaciones y clases parciales que representan las entidades en el modelo conceptual.

⁷ ADO. NET: es un conjunto de clases que exponen servicios de acceso a datos al programador de .Net.

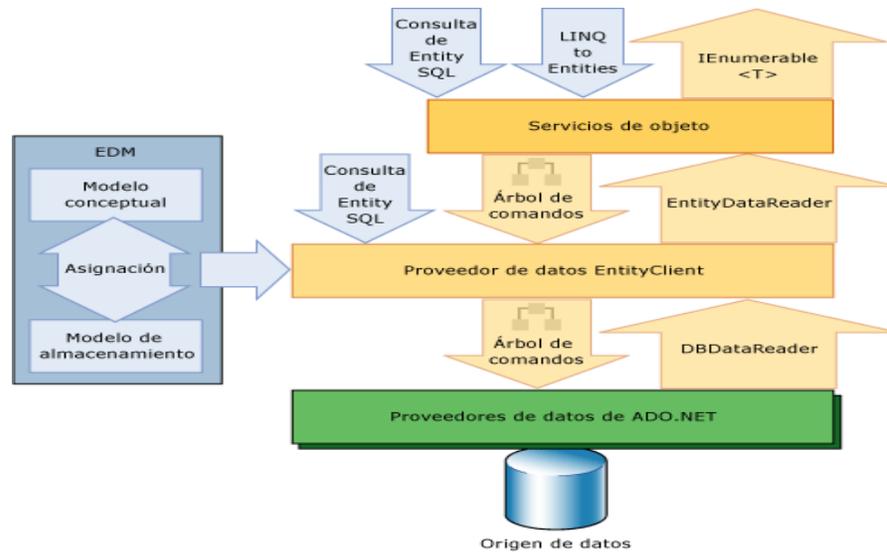


Fig.4 Arquitectura del Entity Framework. (10)

1.9.3 IDE: Microsoft Visual Studio 2008

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, de escritorio y aplicaciones móviles. Los lenguajes de programación que corren sobre este IDE tienen la posibilidad de compartir herramientas y la creación de soluciones. (11)

Existen varias versiones del Visual Studio, una de ellas es la del 2008. La misma se compone por un conjunto global de herramientas que hacen realidad la visión del desarrollador de una forma muy rápida. Se diseñó para admitir proyectos de desarrollo pensados para Web, Windows Vista, Windows Server 2008, Microsoft Office System 2007, SQL Server 2008 y dispositivos de Windows Mobile. Ofrece herramientas de desarrollo avanzadas, funciones de debugging y para bases de datos que permiten crear de manera rápida aplicaciones para distintas plataformas.

Incluye mejoras como diseñadores visuales para un desarrollo más ágil con el Framework .Net 3.5, las herramientas de desarrollo Web y de programación que aceleran el desarrollo a partir de todo tipo de datos. Visual Studio 2008 ofrece funciones de programación y de datos mejoradas, como LINQ (Language Integrated Query), que facilita el armado de soluciones capaces de analizar información y de actuar en consecuencia. Propone ofertas expandidas y mejoradas que ayudan a mejorar la colaboración entre equipos de desarrollo, incluidas herramientas que colaboran con la integración entre profesionales especializados en bases de datos y diseñadores gráficos.

1.9.4 Lenguaje de programación C# 3.0

El lenguaje de programación C# es un lenguaje orientado a objetos que permite a los desarrolladores generar diversas aplicaciones sólidas y seguras que se ejecutan en el Framework .Net. Permite crear aplicaciones cliente para Windows tradicionales, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor y de base de datos.

C# es un lenguaje de programación que ha sufrido mejoras con el paso del tiempo, agregando nuevas e interesantes características en cada una de las versiones que han ido saliendo progresivamente con cada nueva entrega de Visual Studio.

Con la aparición de Visual Studio 2008 surge la versión 3.0 de C# que incorpora nuevas características. Algunas de ellas se mencionan a continuación (12):

- Matrices y variables locales con tipo implícito: indican al compilador que deduzca el tipo de los elementos de variable o matriz en la expresión situada en la parte derecha de la instrucción de inicialización.
- Inicializadores de objetos: permiten asignar valores a los campos o propiedades accesibles de un objeto en el momento de la creación sin tener que invocar explícitamente un constructor.
- Métodos de extensión: extienden de las clases existentes con métodos estáticos que pueden invocarse mediante la sintaxis de método de instancia. Los más comunes son los operadores de consultas LINQ.
- Tipos anónimos: ofrecen un modo útil de encapsular un conjunto de propiedades de sólo lectura en un único objeto sin tener que definir antes un tipo de forma explícita. Habilitan la creación inmediata de tipos estructurados sin nombre que se pueden agregar a colecciones y a los que se puede tener acceso utilizando **var**.
- Expresiones lambda: es una función anónima que puede contener expresiones e instrucciones y se puede utilizar para crear delegados o tipos de árboles de expresión.

1.10 CAGWPFS

Composite Application Guidance for WPF and Silverlight, en español Guía para Aplicaciones Compuestas para WPF y Silverlight, está diseñado con el fin de crear, de manera más fácil, aplicaciones modulares en WPF y Silverlight. Estos tipos de aplicaciones suelen tener múltiples pantallas, flexible interacción con el usuario y la visualización de grandes volúmenes de datos. La vida útil prevista de las mismas se mide en años y cambia en respuesta a nuevas necesidades no previstas. Esta aplicación puede comenzar en pequeño y con el tiempo convertirse en un cliente de aplicaciones compuestas.

Incluye una implementación de referencia, documentación, tutoriales y el código de bibliotecas reutilizables (denominado Composite Application Library) construida sobre el Framework .Net 3.5 y Silverlight. Está destinada para arquitectos y desarrolladores de software.

1.11 Sistema Gestor de Base de Datos Oracle 11g

El Sistema Gestor de Base de Datos Oracle es líder en el mercado por estar en la preferencia de miles de empresas, desarrolladores y administradores de todo el mundo. Oracle 11g es una de las versiones de la empresa Oracle que ha permitido aportar un sistema de administración completo para gestionar todas las situaciones críticas de una base de datos, siendo la escogida por miles de instituciones debido a su escalabilidad y alta disponibilidad.

Oracle 11g está diseñado para entornos de centros de datos en constante cambio y evolución, permitiendo a las empresas adoptar rápidamente nuevas tecnologías mientras se minimiza el riesgo.

Dentro de las características que lo hacen estar en la preferencia de todos se encuentran las siguientes:

- Con el uso de las plataformas grid, proporciona nuevas e innovadoras funcionalidades que garantizan alto rendimiento, alta escalabilidad, fiabilidad y seguridad, asegurando además altos niveles de calidad de servicio e incrementos de la flexibilidad de negocio reduciendo los costes de explotación.
- Haciendo uso de Real Application Testing se reduce de manera drástica los tiempos, riesgos y costes derivados de la implantación de cambios, asegurando que las aplicaciones se comporten de manera adecuada y predecible tras las modificaciones. Además, los clientes ganan en flexibilidad puesto que pueden responder de manera más efectiva a los requerimientos cambiantes del negocio.
- Incorpora Secure Files el cual permite la gestión de todo tipo de datos, incluyendo imágenes, ficheros de texto o tipos avanzados de datos soportados de manera nativa, como XML, imágenes médicas y objetos 3D.
- Incorpora mejoras respecto a otras versiones de Oracle en cuanto a la automatización y la auto-administración de la base de datos permitiendo reducir las tareas de particionamiento y ciclo de vida de los datos lo que proporciona una interfaz simple para visualizar incidencias en la misma.
- Reduce los riesgos con capacidades sofisticadas de seguridad, debido a que existe una mejora en la encriptación de los datos con respecto a otras versiones de Oracle, se integra de manera más estrecha a los módulos de seguridad de hardware para una protección clave de máxima garantía, ofrece entornos adicionales de configuración segura por defecto que soportan políticas de

contraseña y opciones de auditoría y el soporte de autenticación para el administrador y las conexiones del operador del sistema de base de datos es mucho más sólida.

1.12 Patrones

Los patrones son soluciones probadas que permiten construir aplicaciones orientadas a objetos. Sirven de guía a la hora de diseñar sistemas, independientemente del lenguaje en que se implementen y establecen un vocabulario de diseño común entre los desarrolladores. Existen varios tipos de patrones dentro de los que se encuentran:

- **Patrones de Arquitectura:** son aquellos esquemas que permiten la organización de un sistema de software. Se encuentran relacionados a gran escala con el diseño y se aplican específicamente en la fase de elaboración.
- **Patrones de diseño:** son reconocidos también como patrones de micro-arquitectura. Se relacionan con los diseños de objetos y frameworks de pequeña y mediana escala. Como ejemplos de estos patrones se encuentran los GRAPS⁸ y los patrones GOF⁹.

Algunos exponentes de los patrones antes mencionados son los siguientes:

- **Modelo-Vista-Presentador (en lo adelante MVP):** Se deriva del Modelo-Vista-Controlador por lo que su principal función se refleja en la capa de presentación, logrando una estricta regulación de la interacción entre la vista y el presentador. En el MVP la vista y el modelo de datos son separados claramente por medio de una interfaz que muestra la vista, y el presentador tiene acceso de manera polimórfica.
- **Singleton:** Define un método estático en las clases con el fin de lograr que los objetos tengan un único punto de acceso global al admitirse una sola instancia de una clase.
- **Proxy:** Describe maneras comunes de organizar las clases y objetos en un sistema. Permite además la ejecución de un objeto mientras se ejecuta otro.
- **Command:** define una clase por cada acción que implementa una interfaz común. Brinda la posibilidad a los programadores de encapsular la funcionalidad deseada en un objeto reutilizable.

⁸ GRAPS: Patrones de Software para la asignación General de Responsabilidad

⁹ GOF: Gand-of-four("Pandilla de los cuatro")

1.13 Arquitectura de Software

La arquitectura de software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución. Aporta una visión abstracta de alto nivel, mostrando en detalle cada uno de los módulos definidos a pasos posteriores del diseño. Su objetivo fundamental es brindar elementos que ayuden a la toma de decisiones, además de facilitar conceptos y un lenguaje común que permite la comunicación entre el equipo de desarrollo del proyecto.

1.13.1 Arquitectura del Sistema del Centro de Información y Mando.

El Sistema del Centro de Información y Mando contará con una arquitectura base desarrollada en la plataforma Microsoft .Net. Mostrará un diseño modular que facilitará el desarrollo así como las posibles modificaciones o agregaciones que se realicen en el mismo. El estilo arquitectónico elegido para la implementación es el de capas, donde estas acceden a la inmediata inferior a través de interfaces permitiendo que el nivel de acoplamiento entre ellas sea el mínimo posible.

La figura 1 muestra una vista vertical del sistema y todos los componentes que colaboran para su funcionamiento.

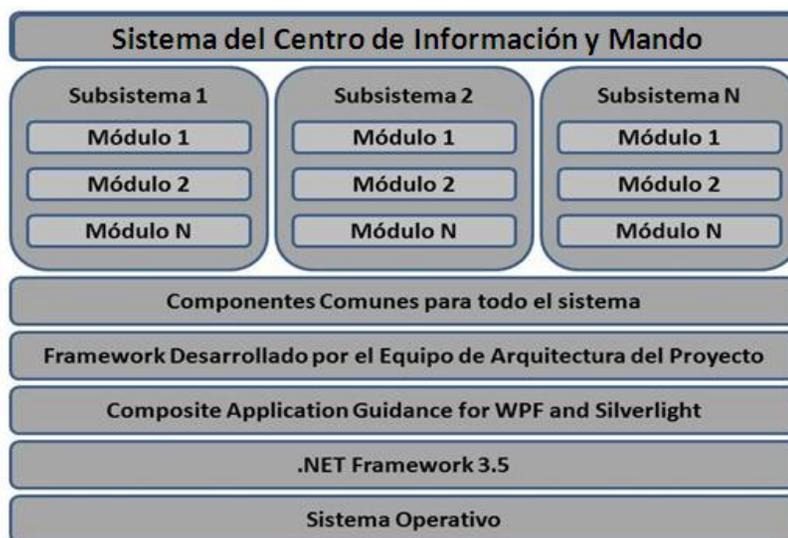


Fig.1 Vista vertical del sistema del Centro de Información y Mando (13)

Como se muestra en la figura anterior el sistema va a estar conformado por subsistemas que estos a su vez van a estar compuestos por módulos o componentes (presentación, acciones, negocio, acceso a datos y dominio). La aplicación va a contar con componentes comunes que unidos a los subsistemas van

a estar dispuestos sobre el framework desarrollado por los arquitectos que conforman el equipo de desarrollo del sistema. Este framework se basa en el Composite Application Guidance for WPF and Silverlight (en lo adelante CAGWPFS) del que se profundizará su función en el epígrafe 1.15. El CAGWPFS estará montado sobre el Framework .Net en su versión 3.5 el cual correrá sobre el sistema operativo Windows XP u otra versión más actual.

1.13.2 Arquitectura de seguridad del Sistema del Centro de Información y Mando

La seguridad informática es un conjunto de soluciones técnicas, métodos y planes que tiene el objetivo de proteger la información manejada en los sistemas informáticos (14). Es así que, priorizar este tema durante la confección de un nuevo sistema, evitando las vulnerabilidades, garantizando que los recursos sean utilizados con los fines provistos y protegiendo los datos y la información que son manipulados por los usuarios de la aplicación resulta de suma importancia.

Dentro de los servicios que se manejan en la seguridad se encuentra el de autenticación de los sistemas informáticos el cual permite verificar la identidad de los usuarios. Se compone por tres elementos fundamentales conocidos como las AAA de seguridad, que se refieren a :

- Autenticación: utilizada para verificar por medio de los mecanismos de seguridad existentes cuáles de los usuarios que intentan acceder a los recursos de la aplicación son quienes dicen ser.
- Autorización: reconocida ya la identidad del usuario solo tendrá acceso a los recursos de la aplicación según los privilegios o permisos que tenga definidos en el sistema.
- Auditoría: una vez que el usuario haya realizado cualquier acción sobre el sistema es necesario entonces mantener un registro sobre las operaciones realizadas con el fin de controlar el acceso y los posibles fallos de seguridad. Este registro se conforma por un conjunto de ficheros donde se trazan las operaciones efectuadas sobre la información o el equipo de trabajo.

La seguridad informática se apoya de mecanismos de gestión de acceso a los sistemas informáticos con el fin de proveer aplicaciones seguras. Uno de estos mecanismos es el basado en roles y permisos de usuarios el cual se desarrolla actualmente por medio del control de acceso conocido como RABC¹⁰.

El RABC es el tipo de control de acceso más usado y el mismo se encarga de administrar la seguridad a través de roles. Estos últimos se asignan a los diferentes tipos de usuarios según sus capacidades y las funciones que realicen. Dentro de los principales términos usados por el modelo RABC se encuentran los

¹⁰ RABC: Control de Acceso a Gran Escala

siguientes: usuarios, roles, permisos, operaciones, objetos y las relaciones que se incluyen en este estándar.

Partiendo de lo anterior se definió que la seguridad del sistema estará basada en permisos, estos serán necesarios para la ejecución de una acción restringida y estarán asociados a los roles definidos por parte de los administradores del sistema. Los roles serán asociados a los diferentes usuarios del sistema para definir los perfiles de acceso a este último.

Este esquema de seguridad basado en permisos facilitará un alto grado de flexibilidad debido a que las maneras de acceso al sistema pueden sufrir varios cambios durante su periodo de explotación. Permite además que ante la necesidad de un cambio en el control de acceso sólo se tengan que realizar modificaciones en la base de datos para definir los permisos de cada rol, así como los roles asociados a cada usuario.

Los usuarios que utilicen el sistema contarán con una cuenta en el mismo y una cuenta única en el Sistema Gestor de Base de Datos (en lo adelante SGBD), con el fin de que si logran saltarse la seguridad de las acciones solo se tenga acceso a los objetos de la base de datos a los cuales están autorizados a acceder. Además, facilitará crear dinámicamente la cadena de conexión al SGBD, de manera tal que no exista la necesidad de tener una cuenta de acceso al gestor de base de datos dentro del código de la aplicación o algún archivo de configuración asociado.

1.14 Conclusiones

Al final de este capítulo se cuenta con una base teórica que fundamenta el desarrollo del Subsistema de Administración y Configuración de Operaciones pues se especifican los conceptos fundamentales de administración, configuración, seguridad, arquitectura de software y centros de emergencias. De estos últimos se describe su situación actual en Cuba y en el mundo, así como los distintos sistemas automatizados que lo soportan. Por otra parte, en este capítulo se detallaron las principales características de la metodología, las tecnologías y las principales herramientas que mejor se adaptan a las necesidades del proyecto. Escogiendo como metodología de desarrollo de software, RUP. Dentro de las razones que avalan esta elección se encuentran las siguientes:

El sistema es de larga duración, de gran complejidad e involucra un gran número de personal y de recursos. La mayoría de los que integran el proyecto son estudiantes de distintos años, lo que trae consigo que en el mismo pueden variar sus integrantes, por tal motivo es de vital importancia contar con la documentación detallada para la continuidad ininterrumpida del proyecto. Además, los clientes residen

fuera de la Universidad de las Ciencias Informáticas lo que imposibilita su permanencia junto al equipo de desarrollo durante la confección del software.

Dentro de las herramientas CASE se escogió el Visual Paradigm para UML debido a que soporta todo el ciclo de desarrollo del software. Además, es una herramienta sencilla que permite la construcción rápida de aplicaciones con una mayor calidad y un menor coste. Facilita la construcción de todo tipo de diagramas, generando documentación y código de los diagramas de clases. Se integra con distintos entornos de desarrollo como el Visual Studio el cual será usado en la construcción del software.

La Modernización del Sistema del Centro de Información y Mando va a estar formado por aplicaciones de escritorio, las cuales deben integrarse y comunicarse entre sí. Todo esto se puede lograr por medio del Framework .Net 3.5 el cual integra tecnologías como el WCF que posibilita una mayor comunicación entre las aplicaciones y WPF que permite la creación de aplicaciones con interfaces más amigables al usuario, además, provee una división entre el diseño de una interfaz y su comportamiento permitiendo a los diseñadores y desarrolladores trabajar de manera independiente.

Se escogió como entorno de desarrollo el Visual Studio 2008 pues el mismo viene junto con el Framework .Net 3.5 y es el IDE que permite aprovechar más a fondo las ventajas de C#. Partiendo de lo anterior se optó por utilizar como lenguaje de programación C# en su versión 3.0, siendo un lenguaje que explota las ventajas de la programación orientada a objetos y de la plataforma en sí. Y por último, con el fin de asegurar una mayor seguridad y consistencia en los datos manejados en el sistema se utilizó como sistema gestor de base de datos Oracle11g, el mismo es líder en el mercado mundial, encontrándose en la preferencia de numerosas empresas y desarrolladores, ya que está diseñado para centros de datos en constante cambio y evolución. Además, ofrece una seguridad sólida ya que reduce los riesgos con capacidades sofisticadas de seguridad, ofreciendo mejoras en la encriptación de los datos.

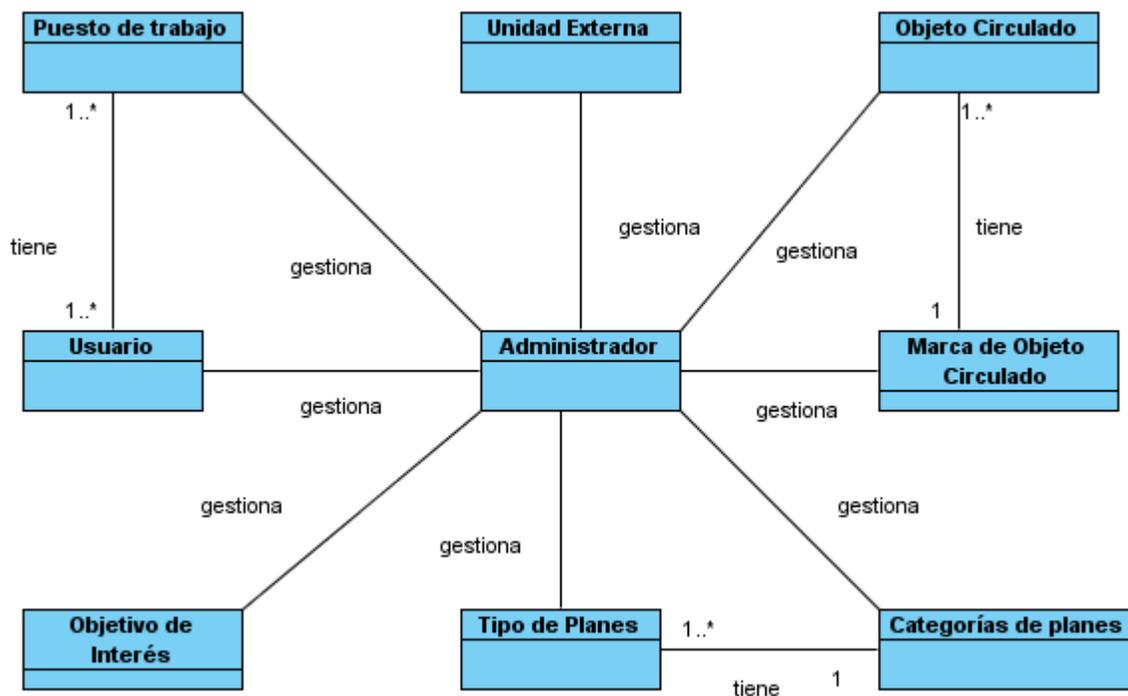
CAPÍTULO 2: CARACTERÍSTICAS DEL SUBSISTEMA

2.1 Introducción

En el desarrollo de este capítulo se especifica el modelo de dominio obtenido como resultado del análisis de los procesos de negocio en la administración y configuración de operaciones. Se enumeran los requerimientos funcionales y no funcionales del sistema describiéndose los casos de usos que dan solución a los mismos. Por otro lado se realiza la propuesta de solución para este subsistema junto con su diagrama de paquetes de Caso de Uso.

2.2 Modelo de Dominio

A partir de los resultados obtenidos en el análisis realizado al negocio del sistema, y los principales procesos del Subsistema de Administración y Configuración de Operaciones, no bien definidos, se realizó un modelo de dominio. En el mismo se describe detalladamente los conceptos claves del negocio así como las relaciones existentes entre ellos.



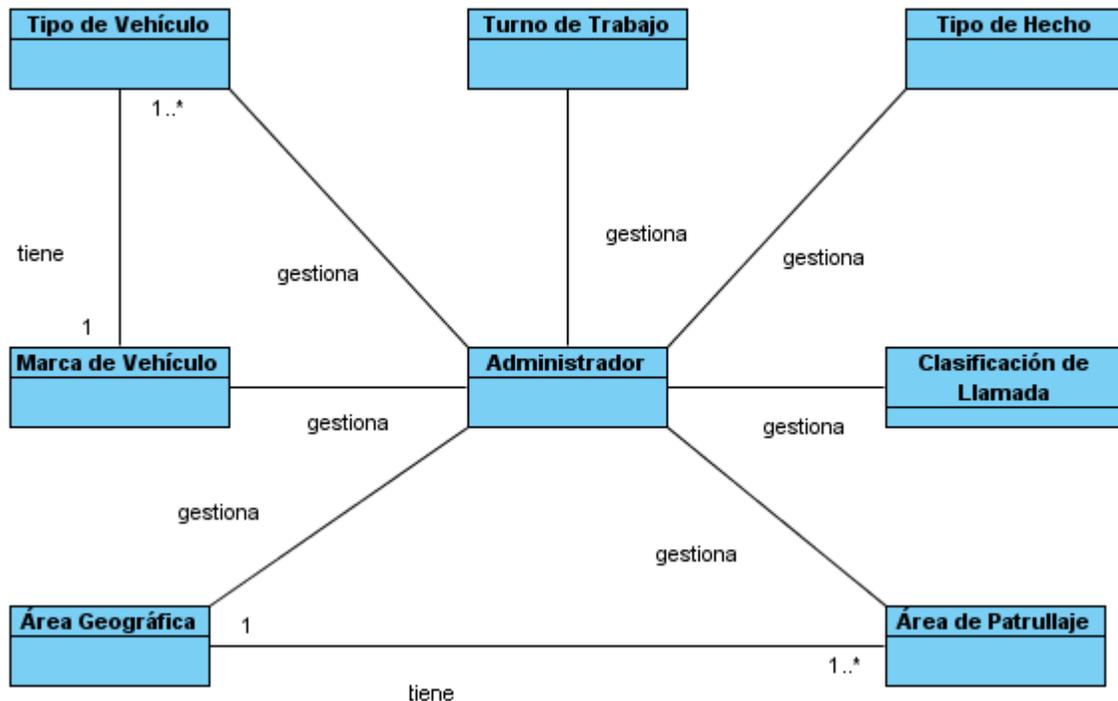


Fig. 4. Modelo de Dominio del Subsistema de Administración y Configuración de Operaciones.

Administrador: Persona encargada de la gestión y configuración del sistema informático.

Usuario: Personas que interactúan con las aplicaciones del sistema.

Puesto de trabajo: Estación de trabajo (computadora) donde se ejecutan y configuran una o varias aplicaciones.

Objetivo de Interés: Área o edificación que se encuentra registrada en el sistema y que es de interés para la toma de decisiones en el Centro de Información y Mando de la UPP. Ejemplo: El Capitolio y La Plaza de la Revolución

Categoría de Plan: Categoría en la que se recoge el tipo de plan.

Tipo de Plan: Maniobra operativa definida por la UPP para enfrentar situaciones especiales. Ejemplo: Cierre del Malecón y Eventos de la Plaza de la Revolución.

Objeto Circulado: Tipo de objeto vinculado a un hecho delictivo. Ejemplo: Aire acondicionado, grabadora y arma de fuego.

Marca de Objeto Circulado: Marca que pueden tener los distintos objetos circulados.

Tipo de Vehículo: Identifica los distintos tipos de vehículos que circulan en la ciudad. Ejemplo: camiones, motos y vehículos ligeros.

Marca de Vehículo: Identifica la marca de los distintos vehículos existentes.

Clasificación de la Llamada: Categoría que se le asigna a las llamadas, en dependencia del motivo por el cual fueron realizadas. Ejemplo: circulación de personas y orden de radio.

Tipo de Hecho: Clasificación de hechos delictivos. Ejemplo: violación, robo y asesinato.

Área Geográfica: Área que comprende un conjunto de zonas geográficas, que no son más que extensiones considerables de terreno cuyos límites están definidos por la UPP a partir de sus necesidades e intereses.

Área de Patrullaje: Área de atención a solicitudes formada por un área geográfica y las patrullas asignadas a las mismas.

Turno de trabajo: Rango de tiempo que demora una sesión de trabajo definida en el horario de los trabajadores del centro.

Unidad Externa: Unidad que se encuentra fuera de los límites geográficos de la UPP. Ejemplo: Puntos de Control.

El administrador del sistema tiene la posibilidad de registrar, modificar y eliminar la información referente a las entidades y nomencladores que se manipulan en el sistema, ellos son: Usuario, Puestos de Trabajo, Objetivo de Interés, Tipos de Planes, Categorías de Planes, Objetos Circulados, Marca de Objeto Circulado, Tipo de Vehículo, Marca de Vehículo, Clasificación de las Llamadas, Tipo de Hecho Delictivo, Área Geográfica, Área de Patrullaje y Unidades Externas. A la hora de registrar un nuevo usuario en el sistema se recogen una serie de datos como: nombre y apellidos, usuario (número del expediente del trabajador) y contraseña. En cada puesto de trabajo se pueden autenticar uno o varios usuarios del sistema, donde estos van a tener acceso a las aplicaciones que se ejecutan en el mismo. El área geográfica establece las zonas geográficas, definidas por la UPP, donde van a operar los carros patrulleros.

Las áreas de patrullaje están conformadas por las áreas geográficas y las patrullas asignadas a las mismas. Los tipos de planes van a estar organizados por categorías, a su vez los tipos de vehículos y los objetos circulados tienen asociados la marca de los mismos. Por cada objetivo de interés se registra el nombre, la dirección donde se encuentra ubicado, el teléfono si es una edificación, los organismos asociados e imágenes del objetivo desde diferentes posiciones. Y por último, se establecen los turnos de trabajo en los que se definen los horarios de los trabajadores del centro.

2.3 Requerimientos Funcionales

Los requerimientos funcionales son cualidades o condiciones que el subsistema debe cumplir, es decir, describen las acciones que el subsistema debe ser capaz de realizar.

A continuación se enumeran los requisitos funcionales que debe cumplir el subsistema a desarrollar:

RF1.Insertar Usuario

Permite adicionar un nuevo usuario a la base de datos a partir de una serie de datos especificados:

1. Nombre
2. Primer Apellido
3. Segundo Apellido
4. Usuario
5. Contraseña
6. Roles
7. Estado de activación

RF1. Modificar Usuario

Permite cambiar los datos de un usuario existente en la base de datos. De los datos entrados no pueden ser modificados el usuario y la contraseña.

RF2. Eliminar Usuario

Permite eliminar un usuario de la base de datos a partir de un usuario seleccionado.

RF3. Buscar Usuario

Realiza una búsqueda de los usuarios, a partir de los siguientes datos:

1. Nombre
2. Primer Apellido
3. Segundo Apellido
4. Usuario
5. Rol
6. Estado de activación

Resultado de la búsqueda:

1. Nombre
2. Primer Apellido
3. Segundo Apellido
4. Usuario
5. Datos biométricos
6. Estado de activación
7. Muestra para el usuario seleccionado el siguiente dato:
 - 7.1 Roles

RF4. Ver Detalles de un Usuario

Permite mostrar todos los datos de un usuario seleccionado.

RF5. Insertar Estado Activación

Permite adicionar un nuevo estado de activación a la base de datos a partir del nombre.

RF6. Modificar Estado Activación

Permite modificar un estado de activación existente en la base de datos.

RF7. Eliminar Estado Activación

Permite eliminar un estado de activación existente en la base de datos a partir de la selección de uno de estos.

RF8. Insertar Rol

Permite adicionar un nuevo rol a la base de datos a partir de:

1. Nombre del rol
2. Permisos
3. Tipo de Autenticación
4. Servicios Web Mapa

RF9. Modificar Rol

Permite cambiar los datos de un rol existente en la base de datos.

RF10. Eliminar Rol

Permite eliminar un rol de la base de datos a partir de la selección de uno de estos.

RF11. Buscar Rol

Realiza una búsqueda de los usuarios, a partir de los siguientes datos:

1. Nombre del rol

Resultado de la búsqueda:

1. Nombre del rol
2. Permisos
3. Tipo de Autenticación
4. Servicios Web Mapa

RF12. Insertar Tipo Autenticación

Permite adicionar un nuevo tipo de autenticación a la base de datos a partir de los siguientes datos:

1. Nombre del tipo de autenticación
2. Descripción

RF13. Modificar Tipo de Autenticación

Permite cambiar los datos de un tipo de autenticación en la base de datos, a partir de la selección de uno de estos.

RF14. Eliminar Tipo de Autenticación

Permite eliminar un tipo de autenticación de la base de datos a partir de la selección de uno de estos.

RF15. Cambiar Contraseña Administrador

Posibilita modificar la contraseña de un usuario registrado por medio del administrador. Los datos necesarios para cambiar contraseña son los siguientes:

1. Usuario
2. Nueva contraseña

RF16. Insertar Puesto de Trabajo

Permite adicionar un nuevo puesto de trabajo a la base de datos con los siguientes datos:

1. Número del puesto de trabajo
2. Número de IP
3. Tipo de Puesto de Trabajo.
4. Estado de activación
5. Subsistemas

RF17. Modificar Puesto de Trabajo

Permite cambiar los datos de un puesto de trabajo a partir de la selección de uno de estos, de los datos el único que no se puede cambiar es el número del puesto de trabajo.

RF18. Eliminar Puesto de Trabajo

Permite eliminar un puesto de trabajo a partir de un puesto seleccionado.

RF19. Buscar Puesto de Trabajo

Realiza una búsqueda de los puestos de trabajo, a partir de los siguientes datos:

1. Número del puesto de trabajo
2. Número IP
3. Tipo de Puesto de Trabajo
4. Estado de activación
5. Subsistema

Resultado de la búsqueda:

1. Número del puesto de trabajo
2. Número IP
3. Tipo de Puesto de Trabajo
4. Estado de activación
5. Muestra para el puesto de trabajo seleccionado los siguientes datos:
 - 5.1 Subsistema

RF20. Ver Detalles de Puesto de Trabajo

Permite mostrar los datos de un puesto de trabajo seleccionado.

RF21. Insertar Tipo de Puesto de Trabajo

Permite adicionar un nuevo tipo de puesto de trabajo a la base de datos a partir del nombre.

RF22. Modificar Tipo de Puesto de Trabajo

Permite cambiar los datos de un tipo de puesto de trabajo a partir de la selección de uno de estos.

RF23. Eliminar Tipo de Puesto de Trabajo

Permite eliminar un tipo de puesto de trabajo a partir de la selección de uno de estos.

RF24. Insertar Área de Patrullaje

Posibilita adicionar una nueva área de patrullaje en la base de datos, para esto se debe insertar los siguientes datos:

1. Nombre del área patrullaje
2. Municipio
3. Consejo Popular
4. Cantidad de patrullas
5. Clasificación

RF25. Eliminar Área de Patrullaje

Permite eliminar un área de patrullaje de la base de datos a partir de la selección de una de estas.

RF26. Buscar Área de Patrullaje

Realiza una búsqueda de las áreas de patrullaje, a partir de los siguientes datos:

1. Nombre del área patrullaje

Resultado de la búsqueda:

1. Nombre del área patrullaje
2. Municipio
3. Consejo Popular
4. Clasificación

RF27. Insertar Categoría Marca

Permite insertar una nueva categoría de marca (objeto, vehículos, armamentos, entre otros) a la base de datos, para esto se deben insertar los siguientes datos:

1. Nombre de la categoría de marca
2. Descripción

RF28. Modificar Categoría Marca

Posibilita modificar los datos de una categoría de marca (objeto, vehículos, armamentos, entre otros) en la base de datos, del cual se pueden modificar:

1. Nombre de la categoría de marca

2. Descripción

RF29. Eliminar Categoría Marca

Permite eliminar una categoría de marca (objeto, vehículos, armamentos, entre otros) de la base de datos a partir de la selección de una de estas.

RF30. Buscar Categoría Marca

Realiza una búsqueda de una categoría de marca (objeto, vehículos, armamentos, entre otros) a partir de los siguientes datos:

1. Nombre de la categoría de marca

Resultado de la búsqueda:

1. Nombre de la categoría de marca
2. Descripción

RF31. Insertar Marca

Permite insertar una nueva marca (objeto, vehículos, armamentos, entre otros) a la base de datos, para esto se deben insertar los siguientes datos:

1. Nombre de la marca
2. Categoría marca
3. Descripción

RF32. Modificar Marca

Posibilita modificar los datos de una marca (objeto, vehículos, armamentos, entre otros) en la base de datos, de la misma se pueden modificar:

1. Nombre de la marca de objeto
2. Descripción

RF33. Eliminar Marca

Permite eliminar una marca (objeto, vehículos, armamentos, entre otros) de la base de datos a partir de la selección de una de estas.

RF34. Buscar Marca

Realiza una búsqueda de una marca (objeto, vehículos, armamentos, entre otros) a partir de los siguientes datos:

1. Nombre de la marca

Resultado de la búsqueda:

1. Nombre de la marca
2. Categoría marca
3. Descripción

RF35. Insertar Modelo

Permite insertar un nuevo modelo (objeto y vehículos) a la base de datos, para esto se deben insertar los siguientes datos:

1. Nombre del modelo
2. Marca
3. Descripción

RF36. Modificar Modelo

Posibilita modificar los datos de un modelo (objeto y vehículos) en la base de datos, del mismo se pueden modificar:

1. Nombre del modelo
2. Descripción

RF37. Eliminar Modelo

Permite eliminar un modelo (objeto y vehículos) de la base de datos a partir de la selección de uno de estos.

RF38. Buscar Modelo

Realiza una búsqueda de un modelo (objeto y vehículos), a partir de los siguientes datos:

1. Nombre del modelo

Resultado de la búsqueda:

1. Nombre del modelo
2. Marca
3. Descripción

2.4 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que especifican las condiciones que el producto de software debe tener para su uso.

A continuación se describen los requisitos no funcionales del subsistema a desarrollar:

RNF1. Usabilidad

- Agrupar botones y vínculos por grupos funcionales

El subsistema deberá presentar grupos de botones y vínculos, organizados por funcionalidad, de tal manera que permita al usuario una interacción consistente con el mismo. Deberá ser de uso intuitivo, de tal forma que se reduzcan los tiempos de entrenamiento, soporte y prueba por parte del usuario. La agrupación de los botones y vínculos por funcionalidad determinará además la capacidad de componer la interfaz de acuerdo a las funciones requeridas para un rol determinado.

- Mensajes y Textos en la Interfaz

Tanto la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma español y tener una apariencia estándar. Los mensajes de error deberán ser lo suficientemente informativos para dar a conocer la severidad del error. Estos mensajes no deben revelar información interna. Ej.: tabla de base de datos o usuario de la base de datos.

RNF2. Fiabilidad

El subsistema debe estar disponible a tiempo completo durante su vida útil.

RNF3. Seguridad

- Almacenamiento de información confidencial

El subsistema deberá permitir el almacenamiento cifrado cuando se trate de información confidencial. Ejemplo: contraseña de usuario, la cual se guardará en la base de datos con un sistema de cifrado irreversible.

- El subsistema debe proteger la integridad de la información y los contenidos

En el subsistema para realizar cualquier operación el usuario debe estar registrado. Todas las acciones que lo requieran deben tener seguridad a nivel de permisos.

- La autenticación se realizará mediante el nombre de usuario y la contraseña

Las contraseñas deberán tener más de siete caracteres de longitud y tener una fortaleza media.

RNF4. Hardware

- Se contará con un servidor con un microprocesador a más de 2.5 GHz, 4GB de memoria RAM y 5GB de capacidad libre del disco duro. Este será empleado para manejar la base de datos junto al servidor de mapa. El sistema operativo a utilizar debe ser de la familia Windows (XP u otra versión más adelantada).
- Los puestos de trabajos deben tener microprocesadores a más de 1 GHz, 512 Mb de memoria RAM.
- Todos estos componentes deben estar interconectados dentro de una misma red de manera que se puedan comunicar.

2.5 Descripción de la Solución Propuesta

La Modernización del Sistema del Centro de Información y Mando de UPP tiene la finalidad de proporcionar un sistema informático con una calidad mayor, y que integre de manera eficiente los subsistemas que lo forman. Con este fin la institución podrá brindar una respuesta más rápida a las situaciones de emergencia que se presenten en la población.

El Subsistema de Administración y Configuración de Operaciones es una aplicación de escritorio a la que solo tienen acceso los usuarios registrados como Administradores. Dentro de las funcionalidades principales con que cuenta el mismo se encuentran gestionar los usuarios y sus privilegios de acceso a las aplicaciones, y la gestión de entidades y nomencladores con el fin de garantizar la consistencia de la información básica del sistema entre los que se encuentran categoría de marca, marca y tipo de puesto de trabajo.

Este subsistema además de la gestión de los puestos de trabajo define las configuraciones de seguridad necesarias para que las aplicaciones puedan ejecutarse correctamente en cada uno de estos. Establece los distintos turnos de trabajo de los trabajadores del centro, así como los parámetros generales para un buen funcionamiento del sistema. Define y visualiza en el mapa las distintas áreas de patrullaje permitiendo la elaboración de las variantes de distribución de los carros patrulleros en cada una de las mismas. Gestiona los distintos procedimientos de trabajo que sirven de guía a los operadores del 106 y oficiales de mando a la hora de dar respuesta a una situación de emergencia determinada o brindarle información de interés a la población. También se registran los diferentes planes operativos definidos por el centro ante eventos importantes y con participación masiva de la población. Y por último, se establecen rangos de evaluación según las categorías de hechos que ocurren en los municipios, permitiendo de esta manera llevar un mejor control de la situación delictiva en los mismos.

2.6 Modelos de Caso de Uso del Subsistema

El modelo de casos de uso expresa las funcionalidades del sistema en términos de casos de usos y la interacción de estos con los actores del sistema. Los casos de uso son fragmentos de funcionalidad que aportan un resultado de valor para los actores, estos últimos son terceros fuera del sistema que interactúan con él.

Como actor del Subsistema de Administración y Configuración de Operaciones se encuentra el Administrador.

Actor	Descripción
Administrador	Persona que interactúa con el Subsistema de Administración y Configuración de Operaciones. Es el responsable de gestionar la información relacionada con los usuarios, puestos de trabajo, parámetros generales, turnos de trabajo, área de patrullaje y otros nomencladores necesarios para el correcto funcionamiento del centro. Realiza

además la configuración de las aplicaciones por puesto de trabajo.

2.6.1 Diagrama de Paquetes

Para lograr una mayor comprensión y claridad se agruparon los distintos casos de uso del subsistema según su funcionalidad en diagramas de paquete como se muestra a continuación:

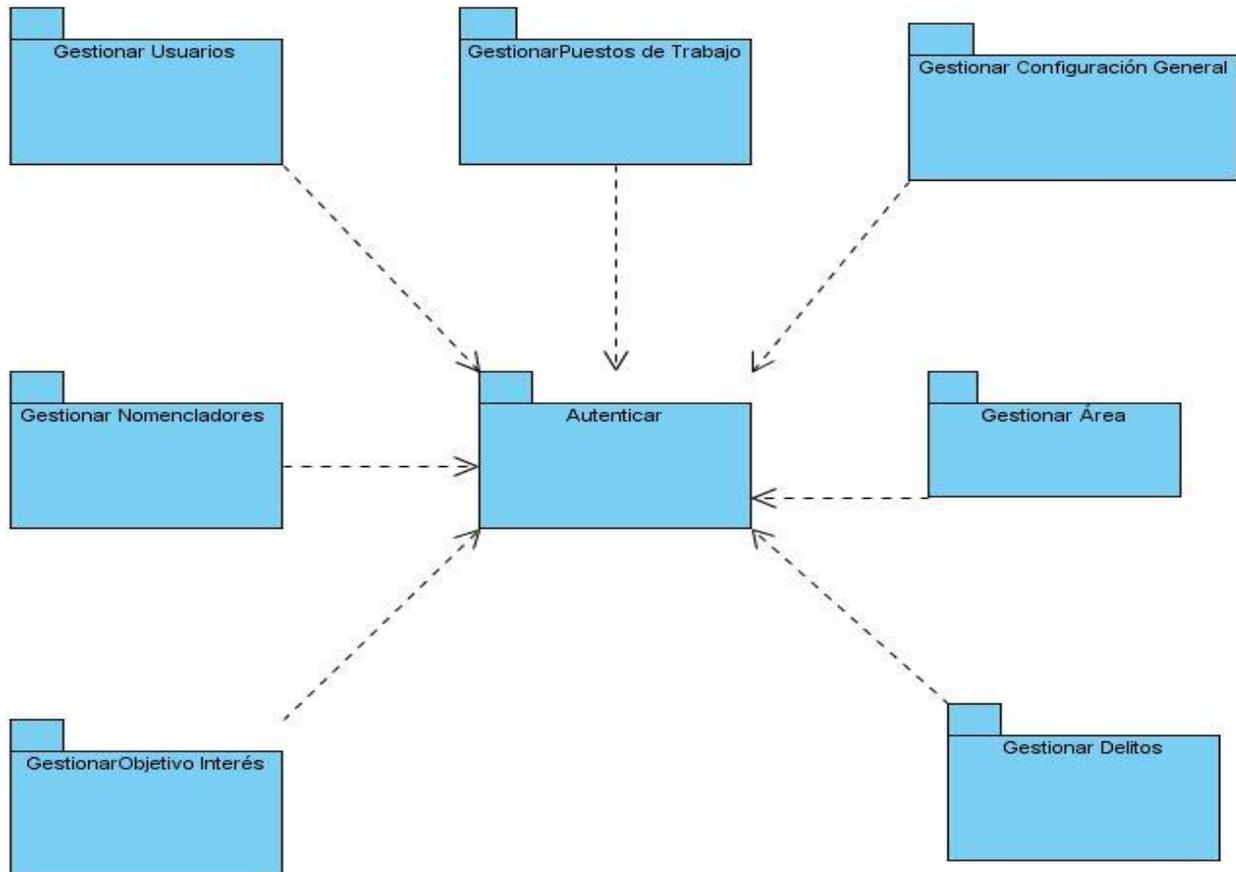


Fig. 4 Diagrama de Paquetes

2.6.2 Diagrama de Caso de Uso

A continuación se muestran los casos de uso agrupados en cada uno de los paquetes:

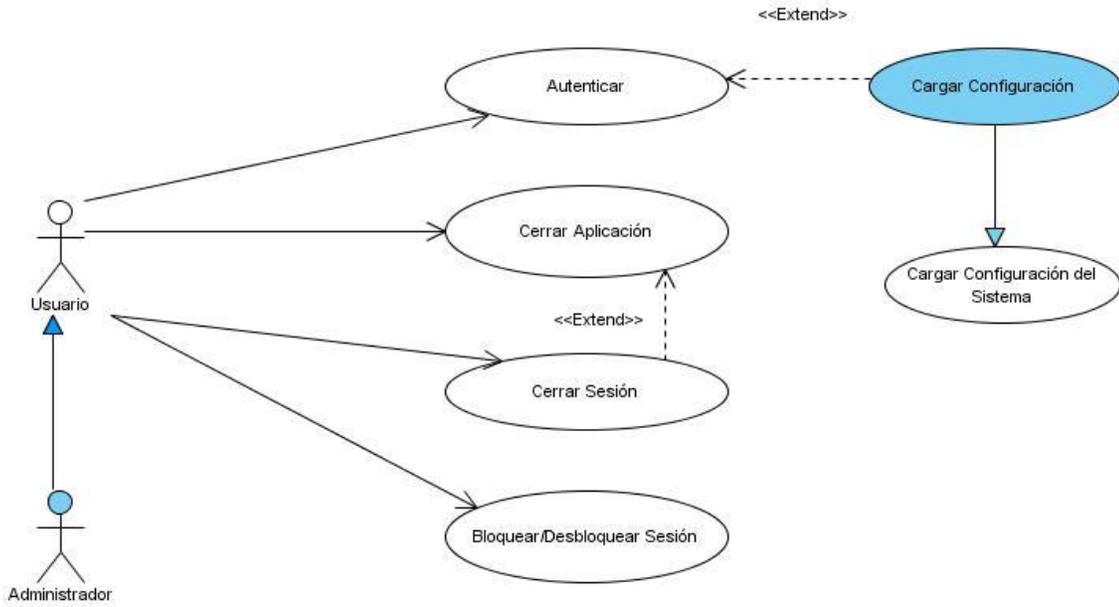


Fig.5 Diagrama de Caso de Uso del Paquete Autenticar

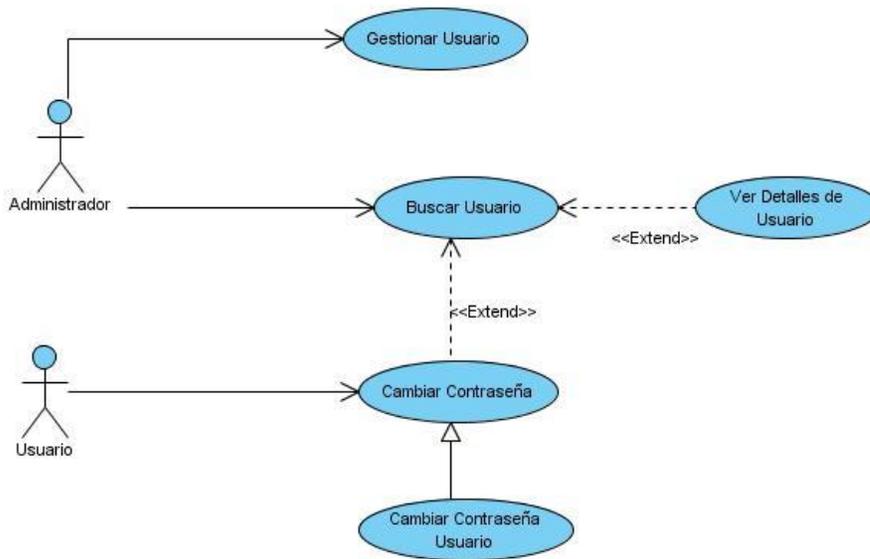


Fig. 6 Diagrama de Caso de Uso del Paquete Usuario

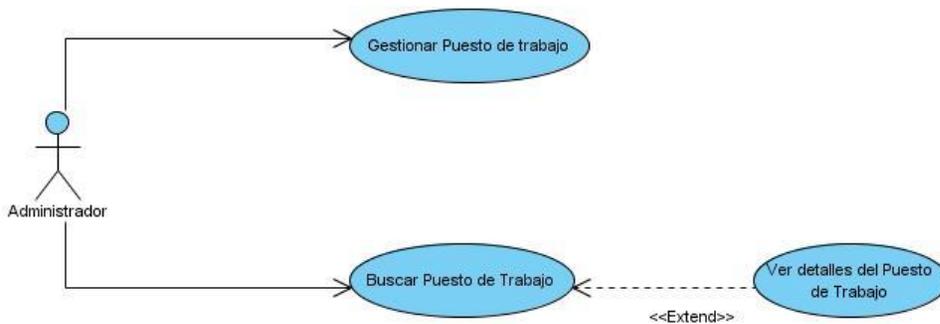


Fig. 7 Diagrama de Caso de Uso del Paquete Puesto de Trabajo



Fig. 8 Diagrama de Caso de Uso del Paquete Área

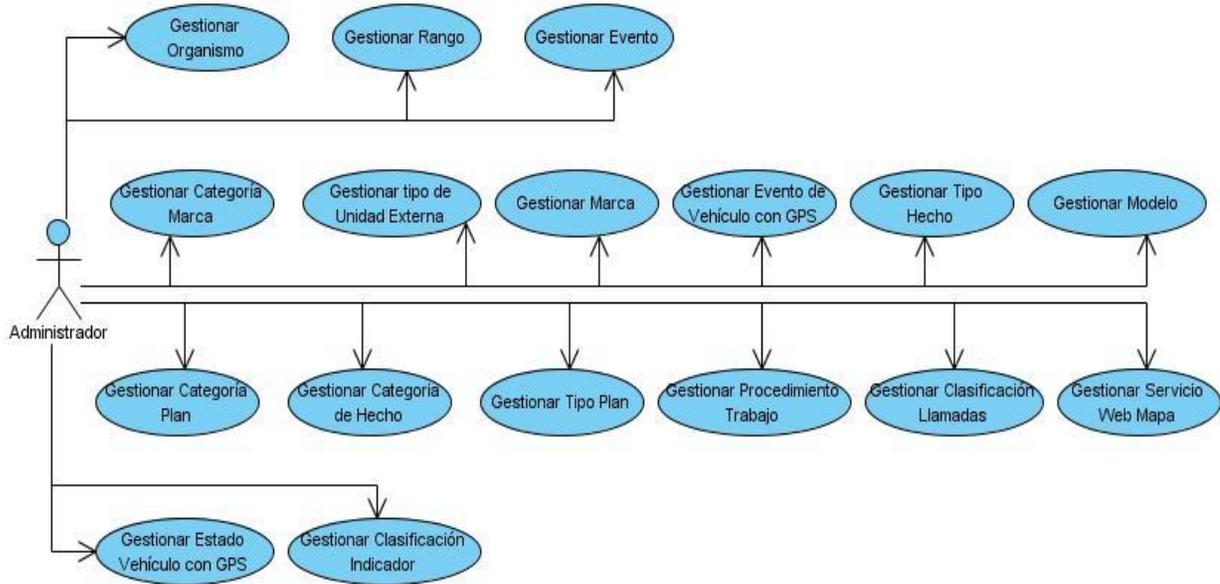


Fig. 9 Diagrama de Caso de Uso del Paquete Nomencladores

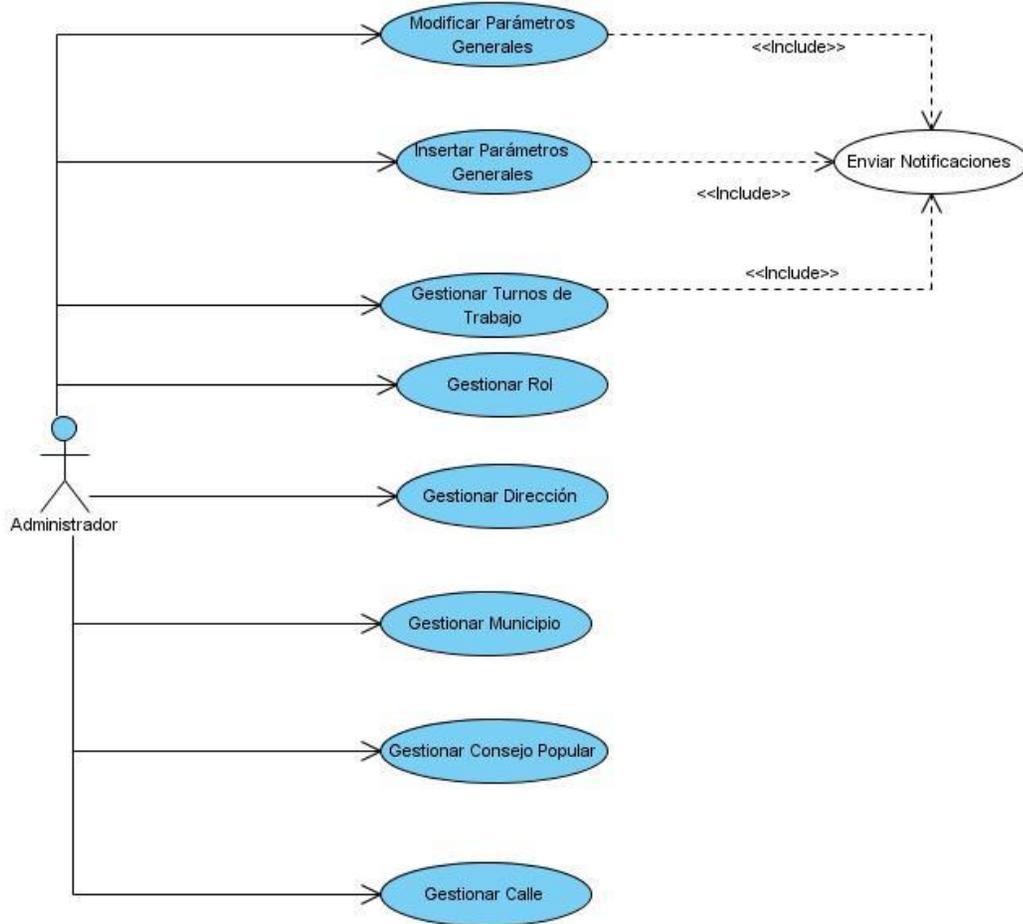


Fig. 10 Diagrama de Caso de Uso del Paquete Configuración General

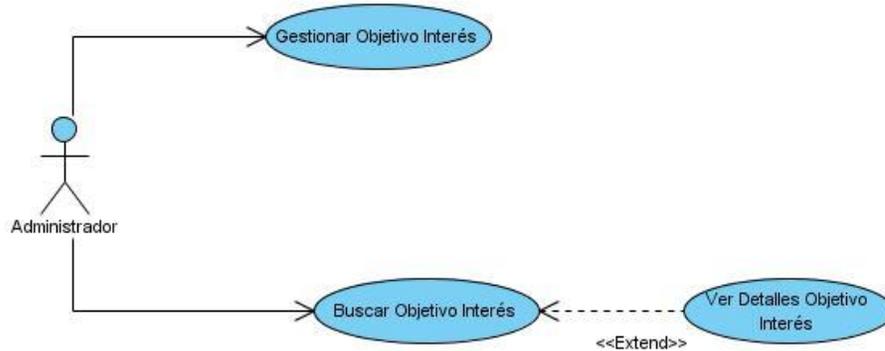


Fig.11 Diagrama de Caso de Uso del paquete Objetivo de Interés

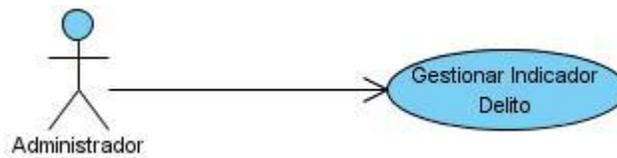


Fig.12 Diagrama de Caso de Uso del paquete Delitos.

Los casos de uso en blanco son los que se utilizan en el subsistema común por lo que no se describen en este documento.

Teniendo en cuenta el significado que tienen algunos de los casos de uso para la arquitectura estos van a tener prioridad 1, y partiendo de que RUP es iterativo e incremental se decidió realizar en una primera versión los siguientes casos de uso:

- CU Gestionar Usuario
- CU Buscar Usuario
- CU Ver Detalles de Usuario
- CU Gestionar Puesto de Trabajo
- CU Gestionar Buscar Puesto de Trabajo
- CU Ver Detalles de Puesto de Trabajo
- CU Gestionar Rol
- CU Gestionar Área de Patrullaje
- CU Gestionar Categoría de Marca
- CU Gestionar Marca
- CU Gestionar Modelo
- CU Gestionar Estado de Activación
- CU Gestionar Tipo Autenticación
- CU Gestionar Tipo de Puesto de Trabajo

2.7 Caso de Uso Expandido

Los casos de uso expandido describen detalladamente la serie de pasos que realiza el actor en su interacción con el sistema.

A continuación se describen los casos de uso del subsistema a desarrollar en la primera versión, la descripción completa de estos y las del resto de los casos de uso se muestran en los **Anexos**.

2.7.1 CU Gestionar Usuario

Caso de Uso	Gestionar Usuario
Actores	Administrador
Resumen	Permite registrar los datos de un nuevo usuario en la Base de Datos, además posibilita modificar o eliminar los datos de un usuario existente en el sistema.
Precondiciones	El administrador debe estar autenticado.
Referencias	RF2,RF3, RF4

2.7.2 CU Buscar Usuario

Caso de Uso	Buscar Usuario
Actores	Administrador
Resumen	Muestra un listado de usuarios que cumplen con los criterios de búsqueda especificados. A partir de la lista mostrada se puede modificar, eliminar, ver detalles o cambiar la contraseña de un usuario existente en el sistema.
Precondiciones	Usuario registrado.
Referencias	RF5

2.7.3 CU Ver Detalles de Usuario

Caso de Uso	Ver Detalles de Usuario
Actores	Administrador
Resumen	Se obtiene de la Base de Datos y se muestra en detalles los datos de un usuario.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF6

2.7.4 Gestionar Rol

Caso de Uso	Gestionar Rol
Actores	Administrador
Resumen	Permite registrar los datos de un nuevo rol en la Base de Datos, además, posibilita buscar los datos de un área de patrullaje existente en el sistema. Muestra un listado de roles que cumplen con los criterios de búsqueda especificados y a partir de estos tiene la posibilidad de modificarlos o eliminarlos.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF10,RF11,RF12,RF13

2.7.5 Gestionar Puesto de Trabajo

Caso de Uso	Gestionar Puesto de Trabajo
Actores	Administrador
Resumen	Permite registrar los datos de un nuevo puesto de trabajo en la Base de Datos, además, posibilita modificar o eliminar los datos de un puesto de trabajo existente en el sistema.
Precondiciones	El Administrador debe estar autenticado
Referencias	RF22,RF23, RF24

2.7.6 Buscar Puesto de Trabajo

Caso de Uso	Buscar Puesto de Trabajo
Actores	Administrador

Resumen	Muestra un listado de puestos de trabajo que cumplen con los criterios de búsqueda especificados. A partir de la lista mostrada se puede modificar, eliminar o ver detalles de un puesto de trabajo existente en el sistema.
Precondiciones	Puesto de trabajo registrado.
Referencias	RF25

2.7.7 Ver Detalles Puesto de Trabajo

Caso de Uso	Ver Detalles de Puesto de Trabajo
Actores	Administrador
Resumen	Se obtiene de la Base de Datos y se muestra en detalles los datos de un puesto de trabajo.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF26

2.7.8 Insertar Área de Patrullaje

Caso de Uso	Insertar Área de Patrullaje
Actores	Administrador
Resumen	Permite registrar los datos de una nueva área de patrullaje en la Base de Datos, además, posibilita buscar los datos de un área de patrullaje existente en el subsistema. Muestra un listado de áreas de patrullaje que cumplen con los criterios de búsqueda especificados y de ahí tiene la facilidad de eliminarlos.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF36,RF37,RF38

2.7.9 CU Gestionar Categoría de Marca

Caso de Uso	Gestionar Categoría de Marca
Actores	Administrador
Resumen	Permite registrar los datos de una nueva categoría de marca en la Base de Datos, además, posibilita buscar los datos de una categoría de marca existente en el subsistema. Muestra un listado de categorías de marcas que cumplen con los criterios de búsqueda especificados permitiendo modificarlos o eliminarlos.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF51, RF52, RF53,RF54

2.7.10 CU Gestionar Marca

Caso de Uso	Gestionar Marca
Actores	Administrador

Resumen	Permite registrar los datos de una nueva marca en la Base de Datos, además, posibilita modificar o eliminar los datos de una marca existente en el subsistema. Muestra un listado de marcas que cumplen con los criterios de búsqueda especificados.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF55, RF56, RF57,RF58

2.7.11 CU Gestionar Modelo

Caso de Uso	Gestionar Modelo
Actores:	Administrador
Resumen	Permite registrar los datos de un nuevo modelo en la Base de Datos, además, posibilita buscar los datos de un modelo existente en el subsistema. Muestra un listado de modelos que cumplen con los criterios de búsqueda especificados permitiendo modificarlos o eliminarlos.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF59, R60, RF61,RF62

2.7.12 CU Gestionar Tipo Autenticación

Caso de Uso	Gestionar Tipo de Autenticación
Actores	Administrador
Resumen	Permite registrar los datos de un nuevo tipo de autenticación en la Base de Datos, además, posibilita buscar los datos de un tipo de autenticación existente en el subsistema. Muestra un listado de tipo de autenticación que cumplen con los criterios de búsqueda especificados permitiendo modificarlos o eliminarlos.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF16, RF17, RF18,RF19

2.7.13 CU Gestionar Estado de Activación

Caso de Uso	Gestionar Estado de Activación
Actores	Administrador
Resumen	Permite registrar los datos de un nuevo estado de activación en la Base de Datos, además, posibilita modificar o eliminar los datos de un estado de activación existente en el subsistema.
Precondiciones	El Administrador debe estar autenticado
Referencias	RF7, RF8, RF9

2.7.14 CU Gestionar Tipo de Puesto de Trabajo

Caso de Uso	Gestionar Tipo Puesto Trabajo
Actores	Administrador
Resumen	Permite registrar los datos de un nuevo tipo de puesto de trabajo en la Base de Datos, además, posibilita modificar o eliminar los datos de un tipo de puesto de trabajo existente en el subsistema.
Precondiciones	El Administrador debe estar autenticado.
Referencias	RF27,RF28,RF29

2.8 Conclusiones

En este capítulo se realizó una descripción del modelo de dominio, el cual representó los conceptos fundamentales en la Administración y Configuración de Operaciones del Sistema del Centro de Información y Mando de UPP, y las relaciones entre estos. A partir del dominio se identificaron los requisitos funcionales que deben estar presentes en esta primera versión del subsistema, los cuales permitirán la creación de un sistema informático capaz de gestionar los datos críticos para el buen funcionamiento de la institución. Por otro lado, se realizó una descripción de los requisitos no funcionales que debe tener el producto de software final. Se mostraron los 8 paquetes que agrupan los diagramas de casos de uso, describiéndose detalladamente las funcionalidades de algunos de los identificados como críticos para la primera versión de la aplicación. Partiendo de todo lo anterior quedaron definidos una serie de elementos fundamentales para identificar qué es lo que el sistema debe hacer, sus límites y las necesidades de interfaz de usuario del subsistema.

CAPÍTULO 3: DISEÑO DEL SUBSISTEMA

3.1 Introducción

En este capítulo se detalla el diseño del Subsistema de Administración y Configuración de Operaciones, mostrando el diagrama de paquetes propuesto para estructurar los componentes del diseño. También se presentan los diagramas de clases del diseño de cada uno de los paquetes y se realiza una descripción de los elementos que componen la arquitectura presente en el subsistema. Por último, se representa el diagrama de clases persistentes y el modelo de datos que va a tener la aplicación en cuestión.

3.2 Arquitectura del Subsistema.

En el capítulo 1 se describió la arquitectura general del sistema, la cual va a estar conformada por varios subsistemas estructurados en capas. En la propuesta de la arquitectura del subsistema de Administración y Configuración de Operaciones se tiene que este va a estar compuesto por la unión de los componentes propios del subsistema y los del mapa, los cuales se comunican de manera directa como se puede apreciar en la figura 13:

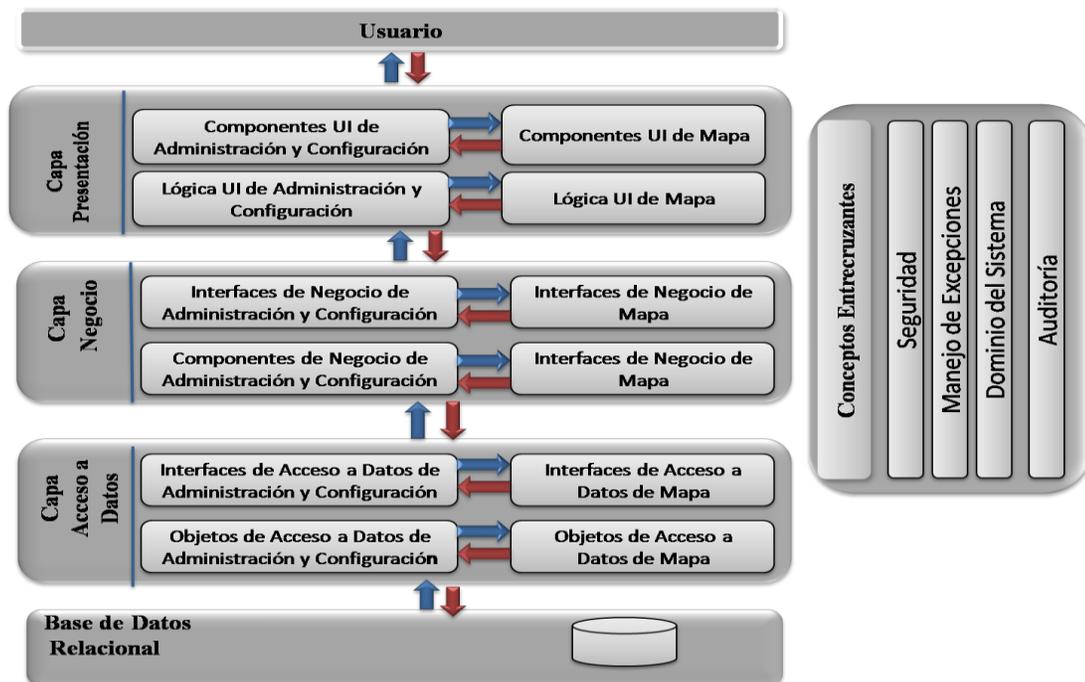


Fig.13 Diseño de la Arquitectura del Subsistema de Administración y Configuración de Operaciones.

Como se muestra en la figura anterior el usuario interactúa con la capa de presentación, la cual es la encargada de definir las interfaces gráficas del subsistema. Esta capa se comunica a través de las acciones (lógica de interfaz de usuario) con las interfaces de la capa de negocio encargada de manejar toda la lógica funcional referente al negocio de la aplicación. Posterior a esto, los componentes del negocio se comunican con las interfaces de la capa de acceso a datos donde se almacena la información en los servidores de base de datos. Es importante aclarar que en todas las capas los componentes del subsistema de administración y configuración de operaciones se comunican directamente con los componentes del mapa correspondientes a su misma capa, con el fin de obtener las funcionalidades que brinda este último.

Dentro del subsistema existen funcionalidades que son transversales, estas son:

- La seguridad se maneja en todas las capas de la solución para evitar accesos no autorizados a la información así como la realización de acciones no permitidas según el nivel de acceso con que cuenta cada usuario.
- El tratamiento de excepciones se realiza en todas las capas en dependencia de lo que se necesite lograr en cada caso.
- El dominio del subsistema está compuesto por todas las clases que son visibles al subsistema, módulos y capas de la aplicación.
- La auditoría se maneja en todas las capas y es la encargada de registrar todas las acciones que se realizan sobre el subsistema.

Explicando lo anterior dentro de un flujo de proceso quedaría de la siguiente manera:

El usuario solicita al subsistema insertar un área de patrullaje, inmediatamente el componente de interfaz de usuario del subsistema se comunica con el del mapa de manera directa y responde a la solicitud mostrando la interfaz que permite adicionar el área, la misma va a contener el mapa donde se podrá visualizar estas áreas. Después de que el usuario inserta los datos en la interfaz, la capa de presentación se comunica a través de la acción `InsertarAreaPatrullajeAccion` con la interface `IInsertarAreaPatrullajeControlador` de la capa de negocio. En esta capa se maneja la clase `InsertarAreaPatrullajeControlador` la cual contiene la lógica funcional referente a la gestión de las áreas de patrullaje. Posterior a esto el componente del negocio se comunica con la interface `IInsertarAreaPatrullajeDao` de la capa de acceso a datos donde se almacena la información en los servidores de base de datos (base de datos donde se almacena los datos del subsistema y la base de

datos donde se recoge los puntos que definen el área en el mapa) por medio de la clase `InsertarAreaPatrullajeDao`.

3.3 Diagrama de Paquetes del Diseño

Los paquetes del diseño son la recopilación de clases, relaciones, diagramas e incluso otros paquetes. Permiten estructurar el modelo de diseño dividiéndolo en partes más pequeñas con fines organizativos y para lograr una mayor compresión de los mismos.

El diseño de este subsistema se dividió en 8 paquetes: Usuario, Puesto de Trabajo, Delitos, Objetivo de Interés, Área de Patrullaje, Configuración, Nomencladores y Autenticar. Para elegir estos paquetes se tuvo en cuenta las relaciones existentes entre las clases contenidas en los mismos.

En el caso del paquete Nomencladores se recogen varios diagramas de clases, dado que la gestión de estos da origen a varias clases, por lo que se hace necesario dividirlo en partes más pequeñas con el fin de lograr una mayor compresión de los elementos contenidos en el mismo.

A continuación se muestra el diagrama de paquetes de diseño del Subsistema de Administración y Configuración de Operaciones:

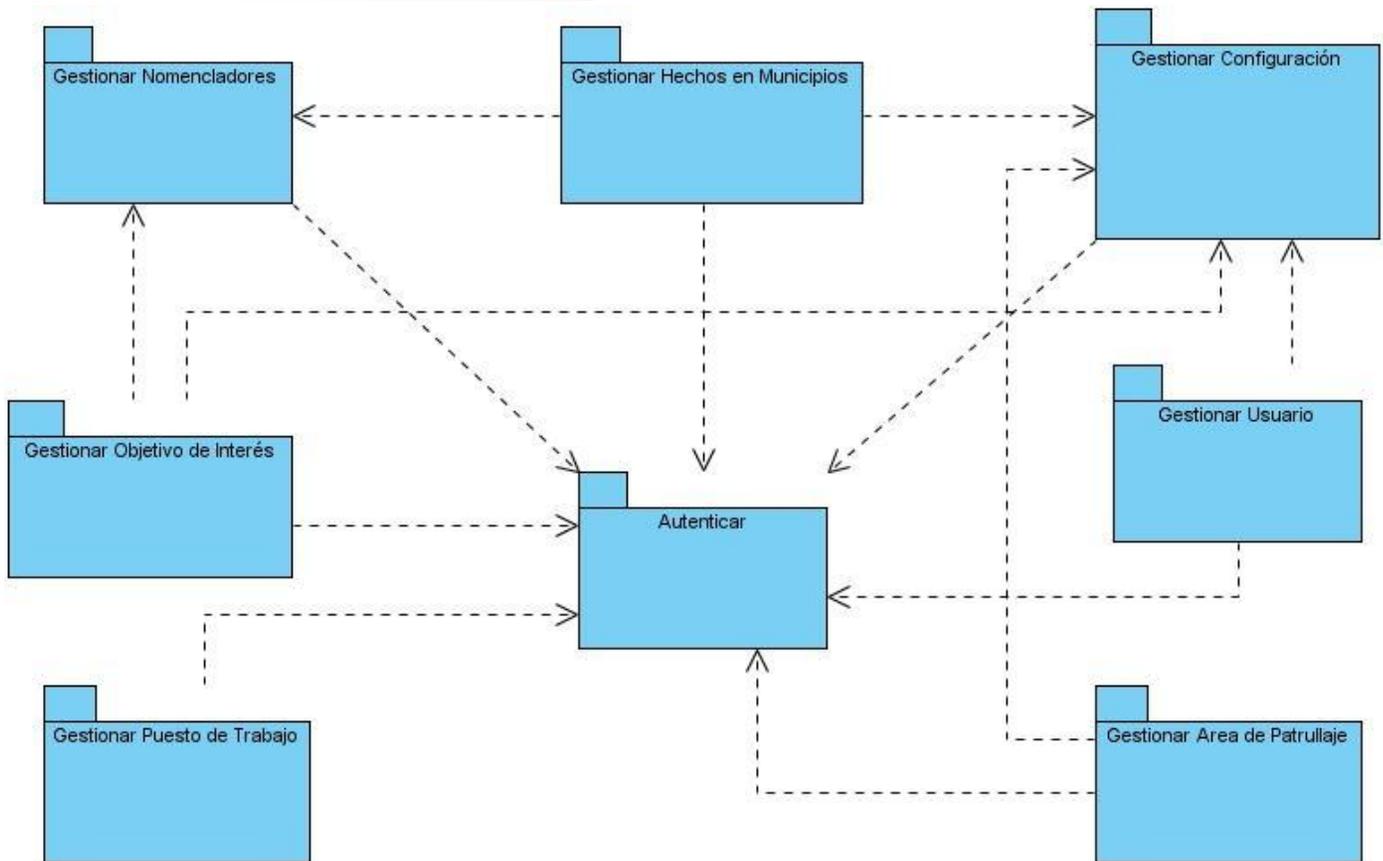


Fig. 14 Diagrama de Paquetes del Diseño.

3.4 Diagrama de Clases de Diseño

Los diagramas de clases se representan como una recopilación de elementos (clases, paquetes y sus relaciones) del modelo declarativo (estático), conectados como gráficos entre si y a su contenido. Se pueden organizar y ser propiedad de paquetes, mostrando solo lo que es relevante en un paquete en concreto. Son el pilar principal del modelado en UML, mostrando tanto lo que el sistema puede hacer (análisis) o como puede ser construido (diseño).

En los diagramas de clase del diseño las clases son la fuerza de trabajo del proyecto de diseño pues las mismas representan una abstracción de una o varias clases en la implementación del sistema, en dependencia del lenguaje de programación utilizado.

En los diagramas de clases de diseño del subsistema, las clases fueron organizadas por colores (ver leyenda en los Anexos) que representan su ubicación dentro de las distintas capas arquitectónicas definidas. En cada uno de los diagramas la clase interfaz de usuario de la capa de presentación se

comunica con la capa de negocio por medio de las acciones. A su vez, la capa de negocio se comunica con la de acceso a datos por medio de la clase interfaz encargada de lograr la comunicación entre ambas capas. En la capa de acceso a datos se recoge también la clase ADAdministracion que tiene la función de mapear las tablas de la base de datos con sus respectivas relaciones. Todas estas clases se relacionan con la clase entidad la cual contiene los objetos que conocen y manejan los datos dentro de la aplicación.

A continuación se muestran algunos de los diagramas de clase de diseño a realizar en la primera versión del Subsistema de Administración y Configuración de Operaciones, el resto pueden ser consultados en los

Anexos:

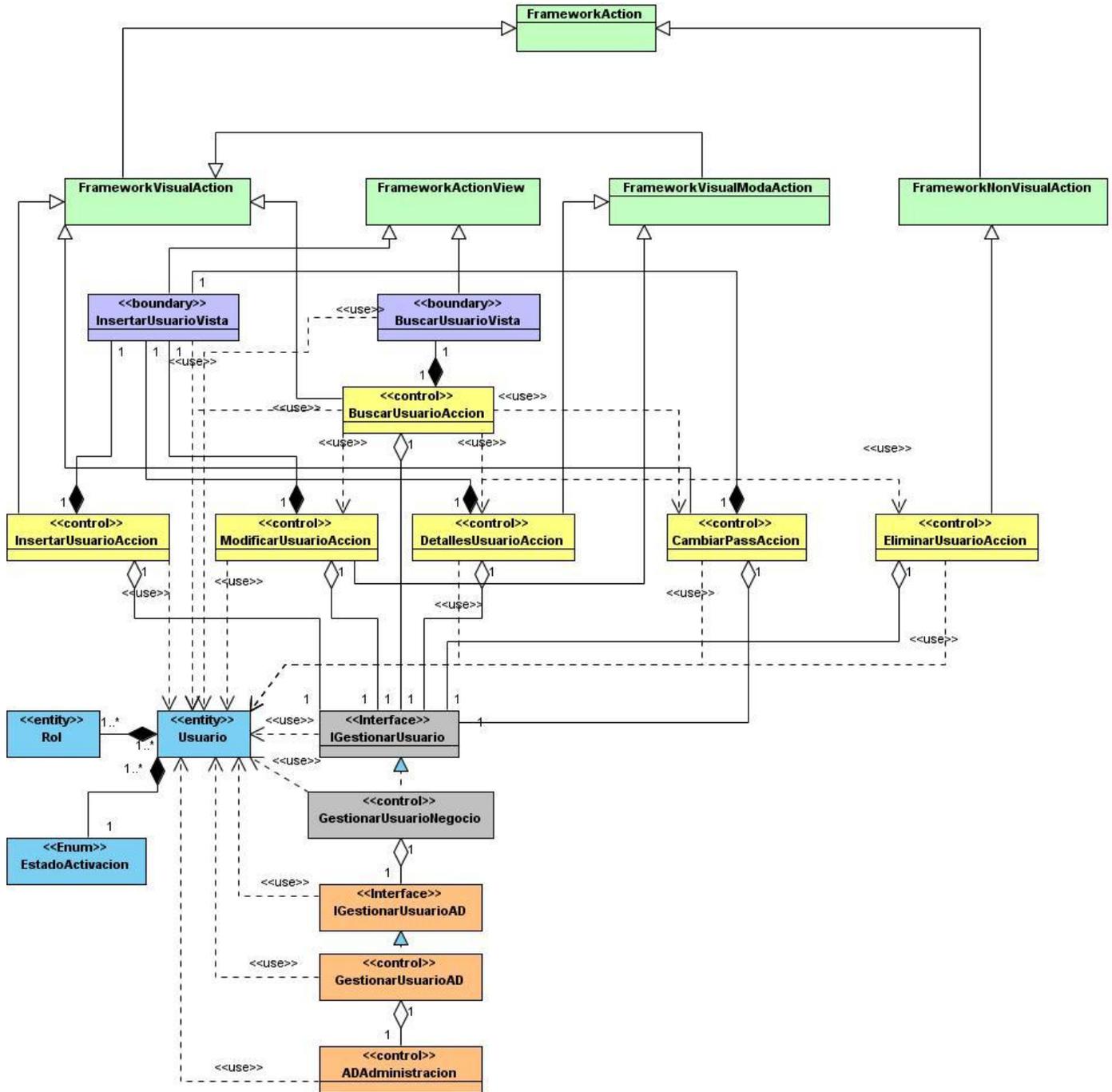


Fig.15 Diagrama de Clases Usuario.

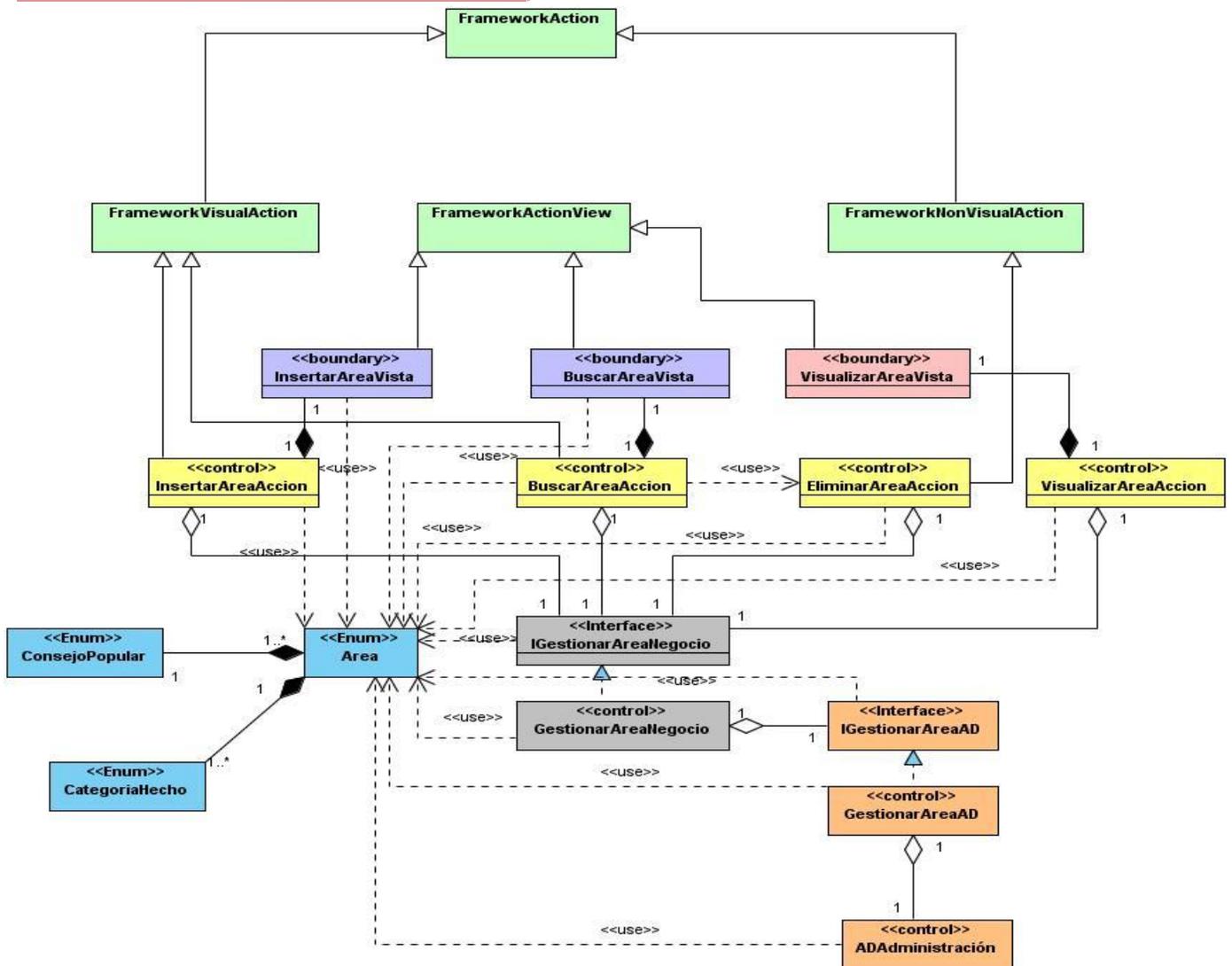


Fig.16 Diagrama de Clases Área de Patrullaje.

3.5 Diagrama de Clases Persistentes

Las clases persistentes son aquellas que deben almacenar su estado en un medio permanente (Base de Datos). La necesidad de recopilar su estado puede estar dada con el fin del registro permanente de la información de las clases, como copia de seguridad en caso de alguna anomalía en el sistema, o para el intercambio de información.

La figura siguiente ilustra el diagrama de clases persistentes del Subsistema de Administración y Configuración de Operaciones.

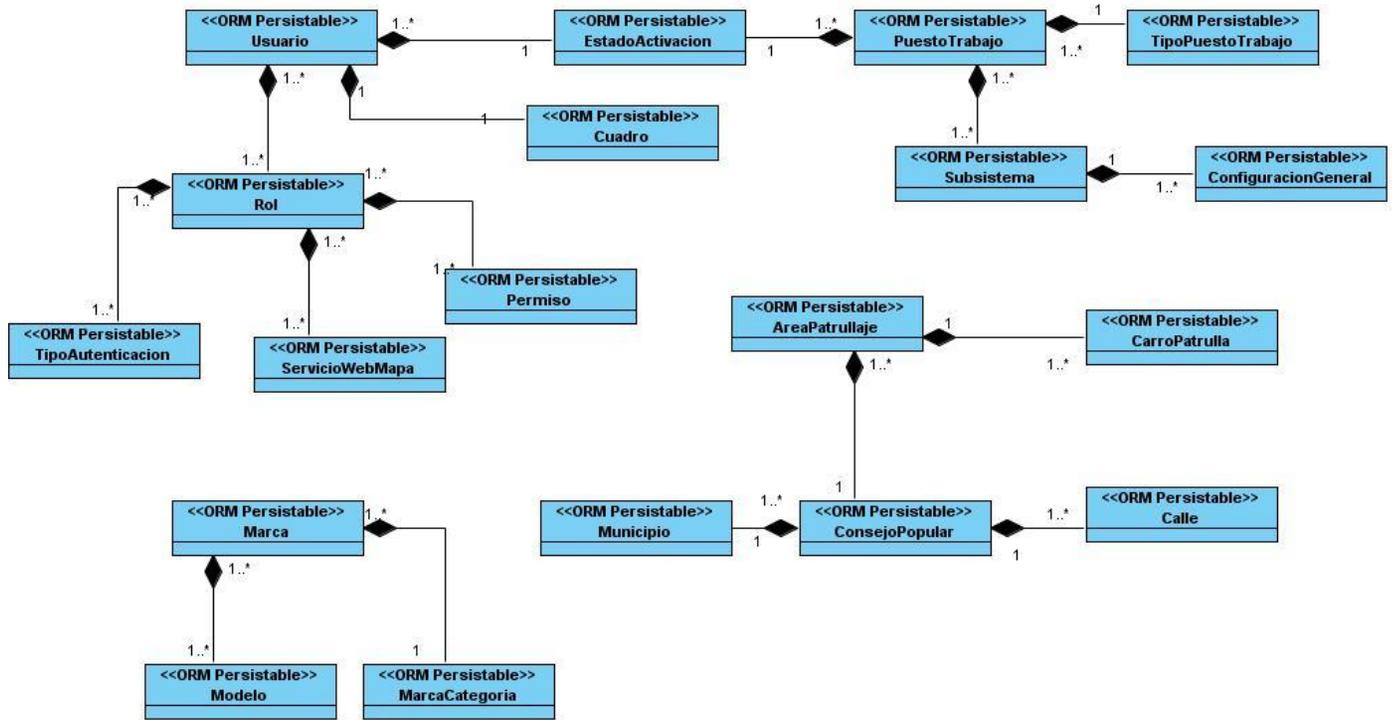


Fig.17 Diagrama de Clases Persistentes

3.6 Modelo de Datos

El Modelo de Datos es utilizado con el fin de describir la estructura lógica y física de la información persistente gestionada por el sistema. Su creación puede partir de la aplicación de ingeniería inversa a la Base de Datos, o por medio de las clases de diseño persistentes obtenidas en el Modelo de Diseño.

A continuación se ilustra el modelo de datos correspondiente al Subsistema de Administración y Configuración de Operaciones:

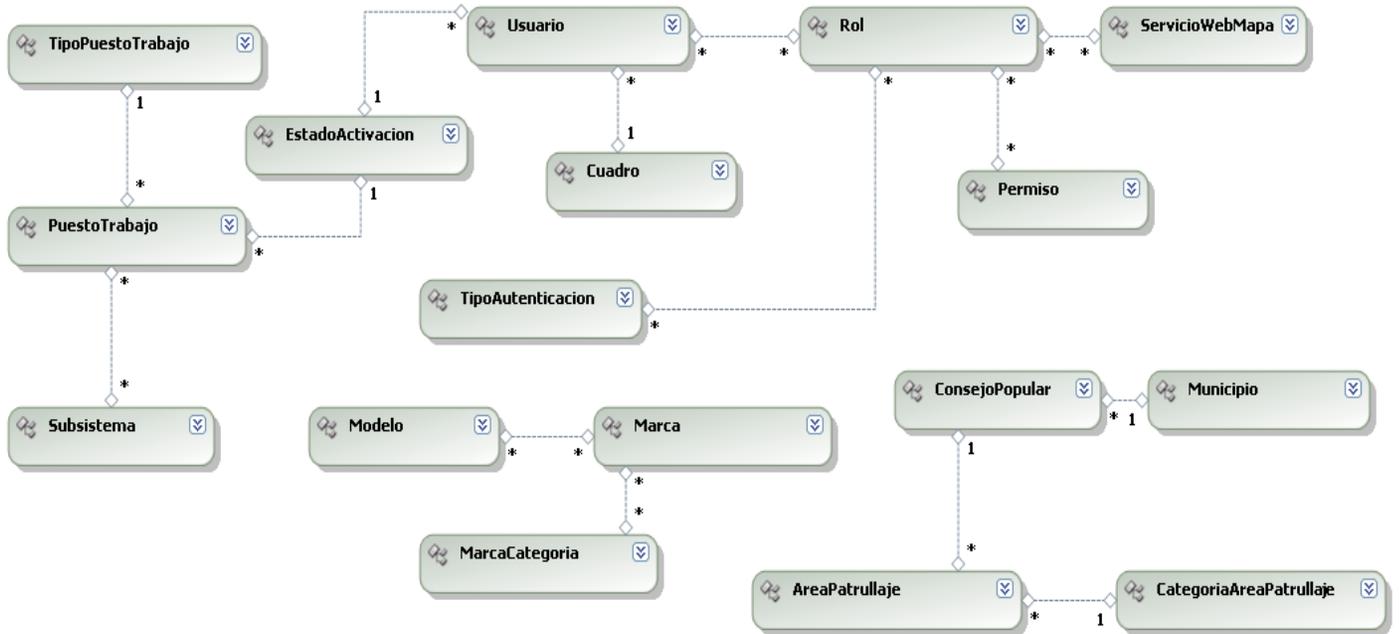


Fig.18 Modelo de Datos

3.7 Conclusiones

En este capítulo se definió el diagrama de paquetes del diseño, el cual tiene la finalidad de agrupar las clases del diseño en dependencia de las entidades que gestionan. Se mostraron los diagramas de clases del diseño correspondiente a cada paquete, además, se describió la arquitectura usada en el desarrollo del subsistema. Por otra parte se mostró el diagrama de clases persistente del subsistema junto con su modelo de datos. Todos estos elementos fueron resultado de la transformación de los requisitos en un diseño del software, que está basado en la arquitectura definida para el sistema, pensado con el fin de obtener el máximo de rendimiento y que constituya la base para la fase de implementación.

CAPÍTULO 4: IMPLEMENTACIÓN DEL SUBSISTEMA

4.1 Introducción

En el presente capítulo se muestra el Modelo de Implementación correspondiente al Subsistema de Administración y Configuración de Operaciones junto con la descripción de cada uno de los componentes relacionados en él. Se muestra además cómo estos componentes se organizan de acuerdo a los nodos del diagrama de despliegue.

4.2 Modelo de Implementación

El modelo de implementación es el encargado de identificar los componentes físicos de la implementación en términos de subsistemas de implementación, y elementos de implementación (directorios y archivos, incluyendo código fuente, datos y archivos ejecutables). (15)

4.2.1 Diagrama de Componentes

Los componentes son la representación de todos los elementos de software (archivos, paquetes y bibliotecas cargadas dinámicamente) que intervienen en la construcción de una aplicación informática, es decir son la parte física y reemplazable de un sistema. El conjunto de componentes unido con sus relaciones conforman el diagrama de componentes. Este es el encargado de modelar los aspectos físicos del sistema, así como brindar una vista de la implementación estática de una aplicación. Su principal uso es representar las dependencias entre ficheros de código fuente, relacionar los elementos de implementación con los del diseño que implementan y mostrar la dependencia entre ficheros en tiempo de ejecución.

A continuación se muestra el diagrama de componentes correspondiente al Subsistema de Administración y Configuración de Operaciones:

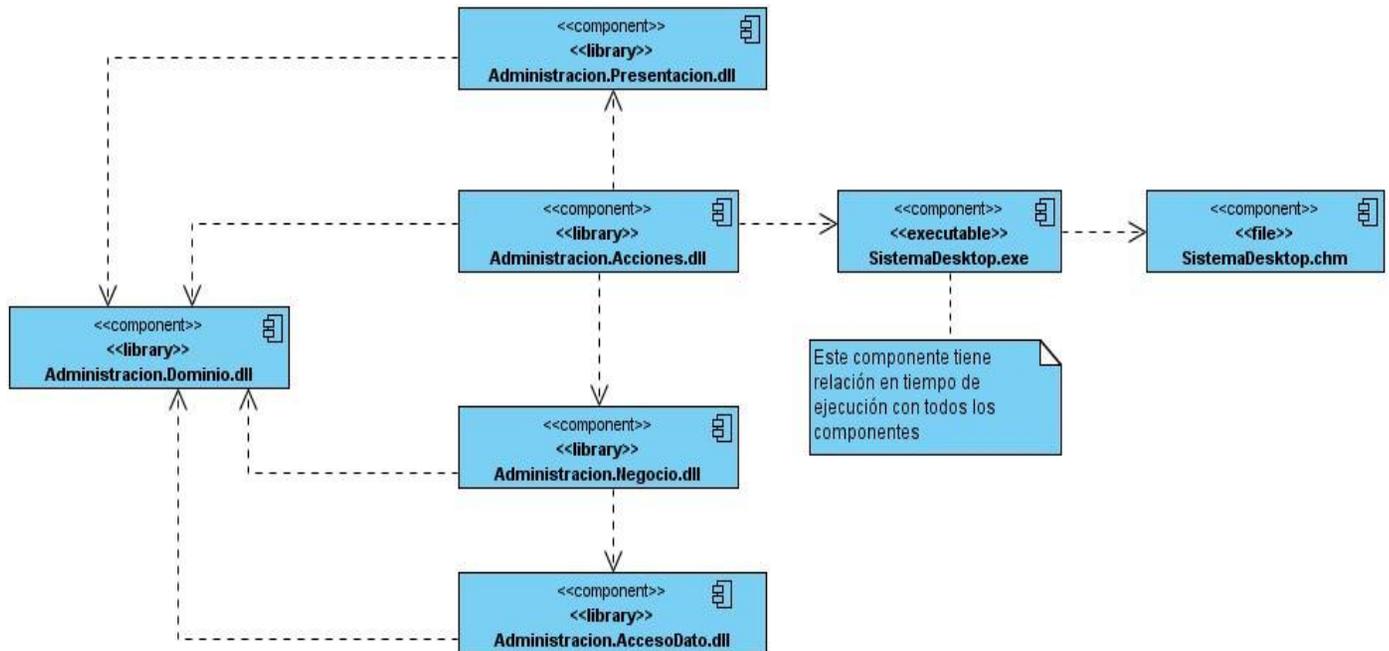


Fig. 20 Diagrama de Componentes

4.3 Descripción de Componentes

A continuación se muestran algunas descripciones de los componentes, el resto puede consultarse en los **Anexos**.

4.3.1 Componente SistemaDesktop.exe

Propósito

Es el encargado de desencadenar la llamada a la acción que da inicio a la ejecución de la aplicación. Contiene los ficheros donde se definen todas las configuraciones generales de la aplicación y el menú.

4.3.2 Componente SistemaDesktop.chm

Propósito

Es la encargada de mostrar el manual de usuario de la aplicación.

4.3.3 Componente Administración.Presentación.dll

Propósito

Contenedor físico de las clases de interfaz gráfica de usuarios. En la aplicación se definen una serie de carpetas según la entidad o el nomenclador que se gestione donde se recogen las interfaces. Ejemplo:

- Administracion.Presentacion. InterfacesUsuario
 - buscarUsuarioVista.xaml
 - InsertarUsuarioVista.xaml
- Administracion.Presentacion. InterfacesAreaPatrullaje

- buscarAreaPatrullajeVista.xaml
- InsertarAreaPatrullajeVista.xaml

4.3.4 Componente Administracion.Acciones

Propósito

Contenedor físico de las clases que realizan un flujo de acción para llevar a cabo una operación determinada. En la aplicación se definen una serie de carpetas según la entidad o el nomenclador que se gestione donde se recogen las acciones que se realizan sobre los mismos. Ejemplo:

- Administracion.Acciones. AreaPatrullajeAcciones
 - buscarAreaPatrullajeAccion.cs
 - eliminarAreaPatrullajeAccion.cs
 - insertarAreaPatrullajeAccion.cs
- Administracion.Acciones. UsuarioAcciones
 - buscarUsuarioAccion.cs
 - eliminarUsuarioAccion.cs
 - InsertarusuarioAccion.cs
 - modificarUsuarioAccion.cs
 - verDetallesUsuarioAccion.cs

4.3.5 Componente Administración.Dominio

Propósito

Contenedor físico de las clases entidades del subsistema.

4.3.6 Componente Administracion.Negocio

Propósito

Contenedor físico de las clases que definen el comportamiento de los objetos de la capa de negocio y de las clases que controlan el negocio de la aplicación. En la aplicación se definen una serie de carpetas según la entidad o el nomenclador que se gestione donde se realiza el negocio de los mismos. Ejemplo:

- Administracion.Negocio. AreaPatrullajeNegocio
 - AreaPatrullajeControlador.cs
 - IAreaPatrullajeControlador.cs
- Administracion.Negocio. UsuarioNegocio
 - IUsuarioControlador.cs
 - UsuarioControlador.cs

4.3.7 Componente Administración.AccesoDatos

Propósito

Contenedor físico de las clases que definen el comportamiento de los objetos de la capa de acceso a datos y de las clases encargadas de gestionar las consultas y modificaciones que se realicen a la base de datos. En la aplicación se definen una serie de carpetas según la entidad o el nomenclador que se gestione donde se realiza el acceso a la información en la base de datos. Ejemplo:

- Administracion.AccesoDatos.AreasDePatrullajeDao
 - IAreasDePatrullajeDao.cs
 - AreasDePatrullajeDao.cs
- Administracion.AccesoDatos.UsuariosDao
 - IUsuariosDao.cs
 - UsuariosDao.cs

4.4 Diagrama de Despliegue

Los diagramas de despliegue representan las relaciones físicas de los nodos que componen un sistema y la organización de los componentes en estos nodos. También se encargan de modelar la vista de despliegue estática del sistema, así como la topología de hardware donde se ejecuta el sistema.

La siguiente figura ilustra el diagrama de despliegue correspondiente al Subsistema de Administración y Configuración de Operaciones:

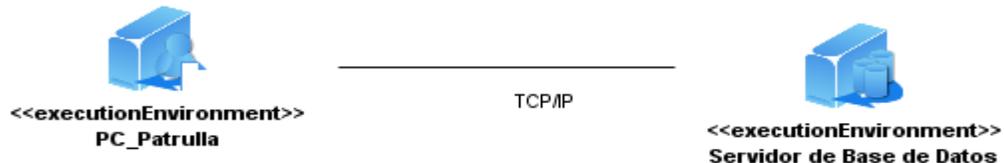


Fig. 21 Diagrama de Despliegue

4.5 Conclusiones

En este capítulo se realizó una representación del modelo de implementación a través del diagrama de componentes, brindando una descripción de los elementos que lo integran como son librerías, ejecutables y la ayuda. Por otra parte se mostró la organización física de los componentes dentro de los nodos definidos en el diagrama de despliegue. Con estos elementos se dio la visión final del subsistema como una aplicación ejecutable, creada a partir de las especificaciones del diseño y que integra los elementos de implementación producidos individualmente.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

5.1 Introducción

En el presente capítulo se realiza un estudio de factibilidad para la realización del Subsistema de Administración y Configuración de Operaciones a partir de la estimación del tamaño, el esfuerzo y de la planificación necesaria para llevar a cabo el mismo. Partiendo de esta estimación y de los beneficios tangibles e intangibles se realiza además un análisis de costo con el fin de probar la factibilidad del subsistema.

5.2 Estimación por Punto de Caso de Uso

Es un método de estimación del esfuerzo de un producto de software teniendo en cuenta los casos de uso presentes en el mismo. Su realización parte de la asignación de pesos a un cierto número de factores que afectan la complejidad técnica y el ambiente en que se desarrolle el producto.

5.2.1 Cálculo de Punto de Caso de Uso sin Ajustar

La siguiente ecuación es usada para calcular los puntos de caso de uso sin ajustar:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

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

Para determinar el Factor de Peso de los Actores sin ajustar (UAW) se debe de tener en cuenta lo siguiente:

Tipo de Actor	Descripción	Factor de Peso	Cant*Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	0	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	0	0*2

Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	1	1*3
Total			3

Fig. 22 Tabla de Valores del UAW

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

No.	Tipo de CU	Nombre del CU	Factor de Peso
1	Medio	Gestionar Usuario	10
2	Medio	Gestionar Rol	10
3	Medio	Gestionar Puesto de Trabajo	10
4	Medio	Gestionar Área de Patrullaje	10
5	Medio	Buscar Usuario	10
6	Medio	Buscar Puesto de Trabajo	10
7	Simple	Ver detalles de Usuario	5
8	Simple	Ver detalles de Puesto de Trabajo	5
9	Simple	Cambiar Contraseña Administrador	5
10	Simple	Gestionar Marca Categoría	5
11	Simple	Gestionar Marca	5
12	Simple	Gestionar Modelo	5
13	Simple	Gestionar Tipo de Autenticación	5
14	Simple	Gestionar Estado de Activación	5
15	Simple	Gestionar Tipo Puesto de Trabajo	5
Total			105

Fig. 23 Tabla de Valores del UUCW.

Partiendo de los resultados de la tabla anterior se tiene que a partir de los 15 CU de ellos 6 presentan complejidad media y 9 complejidad simple.

Luego del análisis anterior tenemos:

$$\text{UUCP} = 3+105$$

$$\text{UUCP} = 108$$

5.2.2 Cálculo de Punto de Caso de Uso Ajustados

A partir de que se tienen los puntos de casos sin ajustar se deben ajustar a partir de la siguiente ecuación:

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor Ambiente

Para determinar el TCF se debe de tener en cuenta lo siguiente:

Este coeficiente se calcula dándole valores a un conjunto de factores que determinan la complejidad técnica del subsistema. Estos valores se cuantifican dándole valores de 0 a 5 según la relevancia que estos tengan. La ecuación utilizada para calcularlo es la siguiente:

$$\text{TCF} = 0.6 + 0.01 \times \sum (\text{Peso} \times \text{Valor asignado})$$

Definición de los valores:

0: Sin influencias

1: Influencia incidental

2: Influencia moderada

3: Influencia media

4: Influencia significativa

5: Fuerte influencia

Factor		Peso	Valor	Σ (Peso*Valor)
T1	Sistema distribuido	2	5	10
T2	Objetivos de performance o tiempo de respuesta	1	2	2
T3	Eficiencia del usuario final	1	1	1
T4	Procesamiento interno complejo	1	0	0
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0.5	2	1.0
T7	Facilidad de uso	0.5	2	1.0

T8	Portabilidad	2	0	0
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	0	0
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	1	0
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	0	0
Total				27

Fig. 24 Tabla de Valores del TCF

Luego del analisis anterior tenemos que:

$$TCF = 0.6 + 0.01 * 27$$

$$TCF = 0.88$$

Para determinar el EF

Constituyen las habilidades y el entrenamiento del equipo de desarrollo de software que inciden directamente en las estimaciones de tiempo. Es un conjunto de factores que se cuantifican en valores de 0 a 5, y se calcula del mismo modo que el factor de complejidad técnica.

La ecuación para calcular el EF es la siguiente:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso} \times \text{Valor asignado})$$

Factor	Descripción	Peso	Valor	Σ (Peso*Valor)
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	5	7.5
E2	Experiencia en la aplicación.	0.5	5	2.5
E3	Experiencia en orientación a objetos.	1	5	5
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	5	10
E7	Personal part-time	-1	0	0

E8	Dificultad del lenguaje de programación.	-1	0	-4
Total				28.5

Fig. 25 Tabla de Valores del EF

Luego del análisis anterior se tiene:

$$EF = 1.4 - 0.03 \cdot 28.5$$

$$EF = 0.425$$

Por tanto:

$$UCP = 108 \cdot 0.88 \cdot 0.425$$

$$UCP = 40.392$$

5.3 Cálculo del Esfuerzo en Horas-Hombre

Con el cálculo del esfuerzo se tiene una estimación más completa de la duración del desarrollo del subsistema. Para ello se debe agregar la estimación del esfuerzo obtenida por los Puntos de Caso de Uso y las del resto de las actividades implicadas con el desarrollo de software.

Para el cálculo del esfuerzo se debe tener en cuenta la siguiente ecuación:

$$E = UCP \times CF$$

Donde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Para determinar el CF se debe de tener en cuenta lo siguiente:

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

$$CF = 28 \text{ horas-hombre (si Total EF} = 3 \text{ ó Total EF} = 4)$$

$$CF = \text{abandonar o cambiar proyecto (si Total FE} \geq 5)$$

A la hora de hallar el total de EF se debe tener en cuenta que:

- Se debe contabilizar cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.
- Se debe contabilizar cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

Según lo anterior se tiene entonces que:

$$\text{Total EF} = \text{Cant EF} < 3 + \text{Cant EF} > 3$$

Total EF= 0

Por tanto:

CF =20 horas-hombre (para un Total EF = 0)

E = 38.522* 20 horas-hombre

E= 807.84

5.3.1 Distribución del Esfuerzo entre las diferentes actividades del Subsistema

Actividad	% del Esfuerzo	Valor del Esfuerzo
Requerimiento	10%	201.96 horas-hombre
Diseño	20%	403.92 horas-hombre
Implementación	40%	807.84 horas-hombre
Prueba	15%	302.94 horas-hombre
Sobrecarga(otras actividades)	15%	302.94 horas-hombre
Total	100%	2019.6 horas-hombre

Fig. 26 Tabla de Valores del Esfuerzo Total

El Esfuerzo Total sería 2019.6 horas-hombre, si se estima teniendo en cuenta las condiciones que presentan los desarrolladores de este subsistema, que un mes tiene 176 horas laborables, pues se trabajan 8 horas diarias 22 días al mes, entonces el Esfuerzo Total en mes-hombre sería 11.475 mes-hombre.

5.3.2 Cálculo del costo del Sistema del Centro de Información y Mando

La siguiente ecuación permite calcular el costo de todo del sistema:

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

Donde:

CH: Cantidad de hombres.

CHM: Costo Hombre - Mes

ET: Esfuerzo Total

Si la cantidad de hombres es 2 y se tiene un Salario Promedio de \$100.00.

Entonces:

$$\text{CHM} = \text{CH} * \text{Salario Promedio}$$

$$\text{CHM} = 2 * 100$$

$$\text{CHM} = 200.00$$

Luego:

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

$$\text{Costo} = 200.00 * 11.475 / 2$$

$$\text{Costo} = \$ 1147.5$$

5.3.3 Cálculo del tiempo de desarrollo del Subsistema de Administración y Configuración de Operaciones

$$\text{Tiempo} = \text{ET} / \text{CH}$$

$$\text{Tiempo} = 11.475 / 2$$

$$\text{Tiempo} = 5.737 \approx 6 \text{ meses}$$

5.4 Beneficios tangibles e intangibles

El Subsistema de Administración y Configuración de Operaciones es un subsistema que se está desarrollando como parte del Sistema del Centro de Información y Mando de la Unidad de Patrullas con el objetivo de tener un control y una organización más estricta sobre todas las personas que interactúan con el sistema. El subsistema tiene entre otros objetivos el de establecer tiempos para garantizar el buen funcionamiento y la eficiencia de la fuerza policial ante cualquier hecho que se presente en la Ciudad de la Habana.

El desarrollo de este subsistema ofrecerá los siguientes beneficios:

- Incorporar nuevas funcionalidades que permitirán influir en la disminución de la duración del proceso dirigido por la Unidad de Patrullas de la Ciudad de la Habana, desde que es recibida la llamada por el Servicio de Telefonistas 106 hasta que el patrullero llega al lugar del hecho.
- El subsistema cuenta con una interfaz agradable y sencilla, mediante la cual el administrador puede introducir los datos que registrarán el buen funcionamiento de la institución así como llevar un control sobre las personas que acceden al sistema.
- El subsistema proporcionará una organización más precisa sobre la información presente en el centro, agrupándola por categorías.

5.5 Análisis de costo

Para determinar si es viable o no el desarrollo del subsistema es importante hacer un análisis de costo que implica la implementación del mismo y los beneficios que aporta, y con ello hacer una estimación de las posibles ganancias.

Partiendo de que la versión 3.5 del Framework .Net es libre y que el cliente paga los costos de las licencias de desarrollo tanto del IDE para .Net como del SGBD de Oracle 11g, sumándole los beneficios que aporta para aumentar la protección de la población ante cualquier tipo de emergencia, se llega a la conclusión de que el desarrollo de este producto es totalmente factible.

5.6 Conclusiones

En este capítulo se realizó el estudio de factibilidad del Subsistema de Administración y Configuración de Operaciones, por medio del método Puntos de Caso de Uso, en el cual teniendo en cuenta el estudio realizado, los beneficios aportados y el costo del mismo se pudo llegar a la conclusión de que el subsistema es completamente factible.

CONCLUSIONES GENERALES

El presente trabajo de diploma que lleva por título Modernización del Sistema del Centro de Información y Mando de Patrulla, Subsistema de Administración y Configuración de Operaciones se ha estructurado en cinco capítulos, que han permitido dar cumplimiento a los objetivos trazados. En estos se caracterizaron las distintas herramientas, las tecnologías, la metodología y la plataforma de desarrollo de software que fueron indispensables en la construcción de la aplicación. Por otra parte la investigación realizada permitió caracterizar los sistemas de gestión de emergencia existentes en el mundo, así como el que se encuentra actualmente en explotación en Ciudad de La Habana, lo que facilitó la comprensión del funcionamiento de estas aplicaciones y determinar propiedades comunes que debían estar presentes en el subsistema. Se identificaron los mecanismos utilizados actualmente en la administración y configuración de aplicaciones informáticas, priorizando los relacionados con la seguridad, la gestión de roles y permisos de usuarios, aportando elementos fundamentales en la implementación. A partir de la identificación de los parámetros de configuración que rigen el funcionamiento del CIMUP, así como los nomencladores y entidades relacionadas con la seguridad y el funcionamiento general del subsistema, se inició el proceso de desarrollo de software el cual originó los artefactos (diagrama de casos de usos, diagrama de clases, diagrama de componentes y diagrama de despliegue) y la documentación correspondiente al mismo. Partiendo de lo anterior se sentaron las bases para el desarrollo del producto final el cual logró solucionar el problema científico que dio origen a la investigación, permitiendo el desarrollo del prototipo funcional del subsistema para administrar y configurar de manera segura el Sistema del Centro de Información y Mando. Además, se realizó un estudio de factibilidad que comprobó que el subsistema en cuestión es completamente factible.

RECOMENDACIONES

Después de concluir con el desarrollo de la investigación se recomienda:

- Continuar el desarrollo del subsistema con el fin de terminar las funcionalidades que restan, así como, refinar las existentes en nuevas versiones.
- Definir nuevos niveles de autenticación como es el caso de la autenticación biométrica.
- Identificar y agregar otros nomencladores cuya gestión sea necesaria.

REFERENCIA BIBLIOGRÁFICA

1. El Prisma. Portal para Investigadores y profesionales. (s.f.). Definición de Administración. [Disponible en:
http://www.elprisma.com/apuntes/administracion_de_empresas/definicionadministracion/]
2. Teltronic. Serie CeCoCo. Soluciones para centros de Coordinación y Control, 2002. [Disponible en:
[http://www.dt410.es/file.aspx?Fil_ID=334\(CECo_SEG\)](http://www.dt410.es/file.aspx?Fil_ID=334(CECo_SEG))]
3. Satdata, 2Mares. SIGEME. Sistema de Gestión de Emergencias. [Citado: febrero 8, 2010], [Disponible en:
http://www.2mares.com/Descargas/Catalogo_SIGEME.pdf]
4. ESRI. El Número Unico de Seguridad y Emergencias de la Ciudad de Medellín, Colombia 1-2-3. [Citado: Mayo, 5, 2010]. [Disponible en:
proceedings.esri.com/library/userconf/latinproc01/seguridad_y_riesgos/articulo-esri-metroseguridad-con%20graficos.pdf]
5. Unidad Provincial de Patrulla. (2000). Manual Técnico. Ciudad de La Habana. [Citado: febrero 5, 2010]
6. Jacobson, I., G. Booch, and J. Rumbaugh, El Proceso Unificado de Desarrollo de Software. Madrid, Addison-Wesley.
7. Ramos, A. C. Herramienta Multimedia de Apoyo a la Enseñanza de la Metodología RUP de Ingeniería de Software, 2009, [Disponible en:
<http://www.eumed.net/libros/2009c/587/Proceso%20Unificado%20de%20Desarrollo%20de%20Software.htm>]
8. Zaragoza, M. d. (n.d.). Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado (RUP). UTM. Órgano Informativo, 2007. [Disponible en:
www.google.com.cu/imgres?imgurl=http://www.utvm.edu.mx/OrganoInformativo/orgJul07/imagenes/RUP1.jpg&imgrefurl=http://www.utvm.edu.mx/OrganoInformativo/orgJul07/RUP.htm&usq=_1JwAvGpDb1S69JMYHopAq2TCbjs=&h=334&w=500&sz=43&hl=es&start=3&itbs=1&tbnid=P-Td4yKLW0MfM:&tbnh=87&tbnw=130&prev=/imagenes%3Fq%3Dimagen%2Bdel%2Bciclo%2Bde%2Bvida%2Bde%2Brup%26hl%3Des%26sa%3DN%26gbv%3D2%26ndsp%3D20%26tbs%3Disch:1]
9. Universidad de Puebla. Herramienta CASE. [Citado: febrero 6, 2010], [Disponible en:
http://www.uppuebla.edu.mx/Profesores/INFORMATICA/PRODUCCION_CIENTIFICA/REBECA/Ingsoftware_II/Apuntes/6.herramicase.pdf]

10. Microsoft. Características de Entity Framework, 2010. [Disponible en: <http://msdn.microsoft.com/es-es/library/bb896338.aspx>]
11. Microsoft. Introducción a Visual Studio. Net, 2010. [Disponible en: [http://msdn.microsoft.com/es-es/library/aa291755\(v=VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa291755(v=VS.71).aspx)]
12. Microsoft. Lo nuevo en la versión de lanzamiento original de Visual C# 2008. Lenguaje C# 3.0, 2008. [Disponible en: [msdn.microsoft.com/es-es/library/bb383815\(VS.90\).aspx](http://msdn.microsoft.com/es-es/library/bb383815(VS.90).aspx)]
13. Vargas, I. Y. Proyecto Técnico del Proyecto Modernización del Sistema del Centro de Información y Mando de Patrulla. Ciudad de La Habana, 2010
14. Orallo, E. H. (s.f.). Seguridad y privacidad en los sistemas informáticos, 2000.
15. Corporation, Rational Software. Ayuda de Rational Unified Process. 2003.
16. Microsoft. Tipos Anónimos, 2010. [Disponible en <http://msdn.microsoft.com/es-es/library/bb384767.aspx>]

BIBLIOGRAFÍA

1. Definición ABC una guía única en la red. Definición de Configuración, 2009. [Disponible en: <http://www.definicionabc.com/tecnologia/configuracion.php>]
2. Microsoft. Descripción de Visual Studio 2008, 2010 [Disponible en: <http://www.microsoft.com/spanish/msdn/latam/visualstudio2008/descripcion.aspx>]
3. Visual Paradigm. Herramienta para el desarrollo de aplicaciones de software, 2010. [Disponible en: <http://www.visual-paradigm.com/product/vpuml/>]
4. Visual Basic. Introducción a la plataforma. Net, 2005. [Disponible en: <http://img.redusers.com/imagenes/libros/lpcu113/capitulogratis.pdf>]
5. Microsoft. Información general y conceptual sobre .Net Framework, 2010. [Disponible en: <http://msdn.microsoft.com/es-es/library/zw4w595w.aspx>]
6. Oracle. Información General sobre Real Application Testing y su Capacidad de Administración, 2007.
7. Morse, C. WPF Tutorial.net. Introduction to Windows Presentation Foundation. [Disponible en: <http://www.wpftutorial.net/WPFIntroduction.html>]
8. Oracle . Oracle Database 11g: Información General sobre las Nuevas Características, 2007. [Disponible en: [http://www.oracle.com/technology/global/lad-es/documentation/collaterals/Oracle-Database11g-New-Features\(cast\).pdf](http://www.oracle.com/technology/global/lad-es/documentation/collaterals/Oracle-Database11g-New-Features(cast).pdf)]
9. Microsoft. Patterns & Practices. Patterns & Practices: Composite WPF and Silverlight, 2009. [Disponible en: <http://compositewpf.codeplex.com/>]
10. SGuía. Redmine, 2009. [Disponible en: <http://www.sg.com.mx/guia/node/527>]
11. Ben Collins-Sussman, B. W. Control de versiones con Subversion, 2002, 2003, 2004. [Disponible en: <http://svnbook.spears.at/nightly/es/svn-ch-1-sect-3.html>]
12. Serrano, L. Oracle Role Manager. Introducción a la administración de Roles, 2008.
13. Microsoft. Ampliación del patrón de MVP para la arquitectura de las UI de aplicaciones de clase empresarial. MSDN Magazine, 2010. [Disponible en: <msdn.microsoft.com/es-es/magazine/ee336019.aspx>]
14. Rojas, M. J. (n.d.). Patrones de Diseños. México: Sistema Nacional de Educación Superior Tecnológica. [Disponible en: <antares.itmorelia.edu.mx/~jcolivar/courses/dp07a/patrones.pdf>]
15. Hyldeé. M, I. N. RBAC: Alternativa Actual para la realización de Control de Accesos a gran escala, **2003.**