

República de Cuba



Universidad de las Ciencias Informáticas

Facultad 2

SISTEMA DE GESTIÓN DE EMERGENCIAS DE SEGURIDAD CIUDADANA (171) “Módulo de Mapificación sobre Web”

Trabajo de Diploma

En opción al título de Ingeniero en Ciencias Informáticas.

Autores: Ailin Alarcón Ferrá

Michel Arias Arias.

Tutor: Lic. Karel Osorio Ramírez.

“Año 49 de la Revolución”
Ciudad de la Habana, Cuba.
2007.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con el mismo.

Para que así conste se firma la presente a los 28 días del mes de junio del año 2007.

Ailin Alarcón Ferrá.

Michel Arias Arias.

Lic. Karel Osorio Ramírez.

Firma del Autor.

Firma del Autor.

Firma del Tutor.

Dedicatoria.

...A nuestros Padres

Agradecimientos

A mis padres, por estar siempre a mi lado, por apoyarme en todo momento, por ser ejemplos de inspiración, por inculcarme el deseo de estudiar, por el cariño, la comprensión y el amor que toda la vida me han dado.

A mi hermano mayor, por haberme ayudado siempre en los momentos más difíciles de mi vida y por siempre darme su apoyo.

A mi hermanito menor, por demostrar que a pesar de su corta edad promete mucho y puede llegar a ser un orgullo para la familia.

A mi familia, que tanto me apoyó en estos años de estudio y sacrificio, por todos los consejos que me dio cuando más los necesitaba.

A mi novia Yuliet, por estar a mi lado, apoyándome en estos momentos, por ser parte de mi corazón y por compartir parte de su vida conmigo, a su familia también por su constante preocupación.

A Norbelis, la madre de mi hermanito, por quererme como un hijo, a su familia también, que me han dado todo su cariño y apoyo.

A mis amigos, que son como mis hermanos, nunca voy a olvidar los momentos que pasamos juntos y a todos aquellos que de una forma u otra siempre se han preocupado por mí.

A Karel, por ser el guía en este trabajo, por su dedicación, apoyo y consagración para que todo saliera lo mejor posible.

A Yaneisy, Adrián, Yordanis, Yisel por ofrecerme sus conocimientos y parte de su tiempo.

A Wilfredo, por apoyarme en la realización de este trabajo.

A mis compañeros de aula y de cuarto, por brindarme su amistad y su comprensión.

A Aymé y a Juan Amador, por su apoyo incondicional.

A mis compañeros del proyecto, por su amistad, apoyo e interés para que todo saliera bien.

A la Universidad de las Ciencias Informáticas, a la Revolución y a Fidel, por darme la oportunidad de ser mejor cada día, por educarme y prepararme ante la vida convirtiéndome en mejor persona.

Michel

Agradecimientos

A mis padres y a Ailime, por siempre estar a mi lado en cada etapa de mi vida, por educarme y guiarme, por todo su comprensión y apoyo.

A mi familia, por el apoyo brindado durante toda mi vida, por enseñarme a salir adelante en los momentos difíciles.

A Elieser, por ser como eres, por estar siempre a mi lado en cada decisión que tomé, por ser mi amigo, por tu apoyo brindado durante todo este tiempo.

A mis amigos, por siempre estar ahí cuando los necesite, por no vacilar en darme su mano ante cada problema, por estar conmigo en los buenos momentos.

A mis compañeros de aula, por los momentos que compartimos juntos, por brindarme su amistad incondicional y su apoyo.

A mis compañeros de proyecto, por el apoyo brindado en la realización de este trabajo.

A la decana, por apoyarme en cada decisión tomada, por guiarme e inculcarme los mejores valores, por educarme y por su comprensión en cada momento.

A Karel, por ser nuestro guía en este trabajo, por la dedicación brindada en cada etapa del mismo.

A la Universidad de las Ciencias Informáticas, a la Revolución y a su líder indiscutible, Fidel, por darme la oportunidad de ser mejor cada día, por educarme y prepararme ante la vida convirtiéndome en mejor persona.

Ailin

Resumen.

Los Centros de Gestión de Emergencias de Seguridad Ciudadana (171) constituyen un servicio que se brinda a la población para dar soluciones a los distintos incidentes que atenten contra su seguridad, y por tanto contra el desarrollo de la sociedad y el bienestar de la población. Estos centros cuentan con un sistema de gestión automatizado que contribuye a que las peticiones formuladas por la población sean atendidas de forma rápida y eficiente.

Hoy en día, en la República Bolivariana de Venezuela no existe un único Centro de Gestión de Emergencias de Seguridad Ciudadana (171), además de que cuentan con un ineficiente sistema automatizado incapaz de integrar a los órganos de seguridad, lo que provoca que la atención a los incidentes formulados por la población no se realice con la rapidez y la eficiencia esperada. Estos sistemas no cuentan con un Sistema de Información Geográfico (SIG), capaz de brindar información al centro y a los Órganos de Seguridad Ciudadana.

Como respuesta a esta situación el Ministro del Poder Popular para las Relaciones Interiores y Justicia, promueve la formulación y puesta en marcha del Sistema de Gestión de Emergencias de Seguridad Ciudadana (171).

Con este trabajo perseguimos el objetivo de realizar el análisis y diseño de un SIG sobre Web como parte del sistema informático del centro que contribuya a mantener informado a los órganos de seguridad ciudadana.

Los SIG constituyen una potente herramienta para el análisis y toma de decisión, posibilitando de esta forma una mejor apreciación de la situación en el territorio y por tanto la adopción de decisiones más precisas.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	4
1.1 Introducción.....	4
1.2 Seguridad Ciudadana.....	4
1.3 Centros de Gestión de Emergencias de Seguridad Ciudadana	4
1.4 Sistema de Información Geográfico (SIG).	6
1.5 Sistemas de Posicionamiento.....	9
1.6 Metodología de desarrollo de software.	10
1.7 Lenguaje de Modelado Unificado (UML).....	13
1.8 Plataformas de Desarrollo.	14
1.9 MapInfo.	20
1.10 Herramientas CASE.	22
1.11 Macromedia Dreamweaver.....	24
1.12 Conclusiones.....	25
Capítulo 2: Descripción de la Solución Propuesta.....	26
2.1 Introducción.....	26
2.2 Modelo de Dominio.....	26
2.3 Requerimientos Funcionales.	29
2.4 Modelo de Caso de Uso (CU).....	37
2.5 Descripción de los Casos de Uso.	41
2.6 Conclusiones.....	80
Capítulo 3: Descripción de la Arquitectura Propuesta.	82
3.1 Introducción.....	82
3.2 Opciones de despliegue.	82
3.3 Vista general de MapXtreme Java Edition.	83
3.4 Biblioteca de etiquetas personalizadas.....	84
3.5 Servlets.	86
3.6 Conclusiones.....	92
Capítulo 4. Diseño del Sistema.....	93

4.1	Introducción.....	93
4.2	Diagrama de Paquetes.....	93
4.3	Diagrama de Clases del Diseño.	94
4.4	Conclusiones.....	107
	Conclusiones	108
	Recomendaciones	109
	Referencias Bibliográficas.....	110
	Bibliografía.....	111

Introducción

La seguridad de los ciudadanos debe ser atendida de acuerdo al mandato establecido en la Constitución de la República Bolivariana de Venezuela, el cual establece: *“Toda persona tiene derecho a la protección del estado, a través de los Órganos de Seguridad Ciudadana regulados por Ley, frente a situaciones que constituyan amenazas, vulnerabilidad o riesgos para la integridad física de las personas, sus propiedades, el disfrute de sus derechos y el cumplimiento de sus deberes”* [1].

Hoy en día, en la República Bolivariana de Venezuela existen varios números telefónicos para reportar emergencias, ofrecidos por empresas privadas de telecomunicaciones, gobernaciones y alcaldías. La mayoría de estos centros solo trabajan en horarios de oficinas y no están estrechamente vinculados. Cuentan con un sistema automatizado incapaz de integrar a los órganos de seguridad, lo que provoca que la atención a los incidentes formulados por la población no se realice con la rapidez y la eficiencia esperada. Estos sistemas, además, no cuentan con un SIG, capaz de brindar información al centro y a los órganos de seguridad ciudadana. Todo esto dificulta la atención con efectividad y rapidez de las emergencias formuladas por la población.

Para darle respuesta a estas situaciones, el Ministro del Poder Popular para las Relaciones Interiores y Justicia, atendiendo a su misión institucional de garantizar la seguridad ciudadana, promueve la formulación y puesta en marcha del Sistema de Gestión de Emergencias de Seguridad Ciudadana (171). Entre los módulos que se desarrollan para este sistema informático se encuentra el desarrollo de un SIG sobre web, que brindará las siguientes funcionalidades:

- Brindar información sobre el servicio de atención a emergencias a los Órganos de Seguridad Ciudadana.
- Visualizar todos los sitios de interés, como por ejemplo: centros hospitalarios, cuerpos de bomberos, centros educacionales, comisarías de la policía, entre otros.
- Mostrar los móviles en servicio y garantizar el seguimiento de las unidades destinadas a un incidente hasta que se logre la asistencia solicitada.
- Visualizar las emergencias que requieran de la atención de una entidad encargada de la seguridad ciudadana.

La Mapificación sobre web surge como necesidad de brindar información sobre el servicio de atención a emergencias a los órganos de seguridad ciudadana de interés a través del mapa digital del área.

El objeto de nuestra investigación es el SIG como apoyo a los Centros de Gestión de Emergencias de Seguridad Ciudadana. Siendo nuestro campo de acción el SIG sobre Web como apoyo a los Centros de Gestión de Emergencias de Seguridad Ciudadana (171) en la República Bolivariana de Venezuela.

Para darle solución a la problemática planteada se definió como objetivo general analizar y diseñar un SIG sobre web como parte del sistema informático del centro. Para dar cumplimiento al mismo se definieron los siguientes objetivos específicos:

- Efectuar sobre el mapa digital, cualquier acción básica, por ejemplo: Acercar, Alejar, Centrar, Mover, Seleccionar, etc.
- Permitir la búsqueda de cualquier elemento referenciado geográficamente.
- Visualizar la ubicación de todos los móviles en servicio y la posición de cualquier sitio de interés.
- Mostrar cada hecho en tiempo real y con carácter histórico.

En el transcurso de nuestra investigación nos planteamos las siguientes interrogantes:

- ¿Cómo dar solución al problema planteado?
- ¿Qué conceptos, tecnologías y software son usados para desarrollar un SIG sobre web?
- ¿Qué metodología utilizar para planificar, gestionar, controlar y evaluar el desarrollo del SIG?
- ¿Qué herramientas y plataforma utilizar para desarrollar la aplicación?

Para dar cumplimiento a los objetivos e interrogantes expuestos se propone:

- Investigar las tendencias conceptuales y de software para desarrollar un SIG sobre web.
- Realizar análisis comparativo de tecnologías utilizadas para el desarrollo de SIG sobre web.
- Seleccionar una metodología para el análisis y diseño de sistemas informáticos, que facilite su creación y garantice su calidad.
- Seleccionar las herramientas idóneas para llevar a cabo el proyecto y elegir la plataforma en la que se desarrollará la aplicación.

El presente documento consta de cuatro capítulos:

- Capítulo 1 denominado “**Fundamentación Teórica**”, se incluyen todos los aspectos teóricos que soportan este proyecto y se analizan algunas de las herramientas y plataformas de desarrollo más utilizadas para la creación de SIG sobre web, justificando las que se utilizarán. Además, se plantea la metodología a seguir en la elaboración del mismo.
- Capítulo 2 denominado “**Descripción de la solución propuesta**”, donde se define el modelo de dominio y sus conceptos fundamentales. Se determinan además los requerimientos funcionales del mismo, agrupándolos en casos de usos.
- Capítulo 3 denominado “**Descripción de la arquitectura propuesta**”, donde se exponen las configuraciones para la arquitectura que propone MapXtreme, realizando un análisis de la configuración de la arquitectura que proponemos para la creación del SIG sobre web.
- Capítulo 4 denominado “**Diseño del Sistema**”, donde se describe la realización de los casos de uso en función de las clases que intervienen y las interacciones entre ellas.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se presenta la fundamentación teórica del trabajo realizado. Se exponen conceptos importantes relacionados con la Seguridad Ciudadana y los Centros de Gestión de Emergencias de Seguridad Ciudadana, prestando una mayor atención a los centros ubicados en la República Bolivariana de Venezuela.

Se describen aspectos relacionados con la investigación, así como se analizan las características que presentan los distintos tipos de SIG y las herramientas que facilitan la realización de los mismos. Se determinará la metodología a utilizar para el análisis y diseño de la aplicación.

1.2 Seguridad Ciudadana

La seguridad es una premisa necesaria para el funcionamiento de la sociedad y uno de los principales criterios para asegurar la calidad de vida de la población. Constituye, además, un derecho de todo ser humano y es la facultad de toda persona a desenvolverse dentro de una sociedad libre de amenazas que atenten contra su vida, integridad física, psíquica o cultural.

El universo de la seguridad ciudadana comprende todo aquello que la amenaza como lo que la protege; de un lado está la violencia, criminalidad nacional e internacional, y del otro lado está el quehacer de todas las instituciones estatales y de la sociedad civil relacionada con la promoción y protección de la misma.

1.3 Centros de Gestión de Emergencias de Seguridad Ciudadana.

Los Centros de Gestión de Emergencias de Seguridad Ciudadana constituyen un servicio que se brinda a la población con el objetivo de ofrecer soluciones a los distintos incidentes que ocurren diariamente en la sociedad, brindando así, una mayor confianza a la seguridad ciudadana de cada individuo.

Para lograr tales expectativas, estos centros deben ser un ente integrador de los organismos de seguridad y emergencia, encargados de recibir las llamadas que se originen desde la población las 24 horas del día durante todo el año, manteniendo un servicio de comunicación que permita garantizar la adecuada supervisión y capacidad de respuesta de los órganos de seguridad.

Estos centros actualmente cuentan con la ayuda de un sistema automatizado que se encarga de realizar la mayoría de las funciones de manera inmediata, este se llama Sistema de Gestión de Emergencias de Seguridad Ciudadana.

Este sistema proporciona las aplicaciones informáticas necesarias para garantizar la gestión automatizada del proceso de atención a emergencias en un centro con estas características. Cuenta con un servicio de recepción de llamadas y otro para el despacho de las emergencias. Permite también la mapificación de la información, tales como: emergencias, móviles en servicio (con Sistema de Posicionamiento Global), etc. Incluye, además, herramientas de apoyo para la generación de estadísticas y supervisión del proceso.

Tiene la finalidad de disminuir en gran medida el tiempo de respuesta a las demandas formuladas por la población, brindando un mejor servicio y contribuyendo de una forma u otra a que los ejecutivos tomen mejores decisiones para el bienestar y la seguridad del pueblo.

En nuestro país se han dado los primeros pasos para la creación de centros como este, actualmente en Ciudad de la Habana, los ciudadanos pueden recurrir para informar cualquier incidente al número 106, de la Policía Nacional Revolucionaria. En esta provincia también existe el número 105, de los Bomberos. Independientemente de que existan dos números telefónicos para la atención de las emergencias el número más usado actualmente es el 106.

En el mundo hay otros países que también han creado centros con estas características, teniendo en cuenta las necesidades de cada región y los recursos que tienen a su disposición; tal es el caso del 911 en Estados Unidos y Argentina, el 112 que permite a cualquier ciudadano que reside en la comunidad de la Unión Europea realizar llamadas a los servicios de urgencia y el 123 que se emplea actualmente en Colombia.

1.3.1 Centros de Gestión de Emergencias de Seguridad Ciudadana (171).

En la República Bolivariana de Venezuela, no cuentan con un único sistema para la gestión de emergencias. Existen algunos estados con centros como este, pero no con las condiciones y características requeridas para cubrir las necesidades de la población.

La mayoría sólo responde en horas de oficina y no están conectados entre sí, por lo que actúan como centros independientes. Tampoco presentan un sistema que visualice las demandas formuladas por la población en un mapa digital del área, que permita mantener informado a los Órganos de Seguridad Ciudadana.

A continuación se mencionan algunos de los centros que funcionan actualmente en Venezuela, con los números de teléfono donde se pueden encontrar:

- Sistema Integral de Emergencia de Táchira (171).

- Servicio Autónomo de Emergencias del Estado Bolívar (171).
- Servicio Autónomo de Emergencias de Aragua (171).
- Alcaldía Metropolitana (864-7191).
- Alcaldía Libertador (545-4513 / 542-1711 / 409-8632).
- Alcaldía Sucre (237-6343 / 271-0253 / 272-3360 / 0-800-76547).

Los centros que actualmente funcionan se encuentran incapacitados para integrar y mantener informados a los Órganos de Seguridad Ciudadana, demorando su respuesta y por lo tanto, resultan ineficientes a la hora de brindar una solución. Además de que no cuentan con un SIG sobre web que contribuya a erradicar las deficiencias planteadas anteriormente. Todo esto repercute en el bienestar, la confianza y seguridad de la población venezolana.

1.4 Sistema de Información Geográfico (SIG).

Los SIG han constituido durante los últimos años una potente herramienta de trabajo de uso masivo para la creación y gestión de información espacial. Un SIG puede definirse como un sistema de información capaz de integrar, almacenar, editar, analizar y mostrar en pantalla información referenciada geográficamente.

Consta principalmente de 4 componentes: software, hardware, base de datos y personal.

- Hardware: Constituyen todos los dispositivos sobre el cual opera el SIG.
- Software: Proveen las herramientas y funcionalidades necesarias para almacenar, analizar y presentar información geográfica, los componentes principales del software son:
 - Herramientas para la entrada y manipulación de la información geográfica.
 - Herramientas que soportan consultas, análisis y visualización de informaciones geográficas.
 - Sistema manejador de base de datos geográficos.
 - Una interfaz gráfica amigable para el usuario y de fácil acceso a las herramientas de trabajo.
- Bases de Datos: Las bases de datos en los SIG son llamadas bases de datos geográficas. Ofrecen las mismas funcionalidades que las bases de datos pero además soportan un nuevo tipo de datos para trabajar con los datos geométricos. Estas son la esencia del sistema y no es más que una colección de datos acerca de objetos localizados en un área determinada. De igual forma se permite acceder a información geográfica que esté ubicada fuera de una base de datos, en

estos casos la información está distribuida en ficheros, cuando son accedidos desde el sistema su tratamiento es igual al empleado que cuando se accede desde una base de datos, esto es debido a que el almacenamiento interno de estos ficheros responde, analógicamente, a la estructura de las tablas de un Modelo Relacional.

- Personal: Es el encargado de operar, desarrollar y administrar el sistema. Además, realiza análisis complejos siguiendo los criterios impuestos en la realización del Software y de la toma de decisiones.

El SIG es una herramienta de gran importancia en el análisis y la toma de decisiones. Aporta soluciones a diferentes problemas que frecuentemente requieren de un rápido acceso a información de varios tipos, que solo pueden estar relacionadas por geografía o distribución espacial.

SIG Vectorial

Los SIG vectoriales son aquellos que para la descripción de los objetos geográficos utilizan vectores.

En este modelo la información se representa a través de puntos, líneas y polígonos almacenada como una colección de coordenadas (x, y). A estos objetos de dibujo ya se les puede asociar las diversas capas de información que se relacionan con el modelo espacial generado a través de puntos y líneas.

- Los puntos se reducen a pares de coordenadas latitud-longitud o x-y, que marcan la posición de lo que es modelado sobre la superficie de la tierra. Así, los pozos, fuentes, manantiales, puntos contaminados pueden quedar representados con esta estructura vectorial.
- Las líneas son una serie ordenada de posiciones unidas por segmentos rectos. Permiten modelar carreteras, ríos, curvas de nivel.
- Los polígonos son líneas cerradas que delimitan superficies. Son empleados para modelar vegetaciones, suelos, geologías, montes, provincias, países.

Este tipo de SIG es muy útil para describir características discretas, pero menos útil para describir características de variación continua.

SIG Raster.

Utilizan una malla rectangular de pequeñas celdas, denominadas píxel. Cada píxel recibe un número como representación a su valor temático. Este número porta la información necesaria para modelar un aspecto del medio. Dado que la malla es regular (el tamaño del píxel es constante) y que conocemos la posición en coordenadas del centro de una de las celdas, se puede decir que todos los píxeles están georreferenciados.

El modelo de datos raster es especialmente útil cuando tenemos que describir objetos geográficos con límites difusos o de medios muy variables.

SIG Orientado a Objetos.

Es un modelo de datos que ha surgido en estos últimos años. Plantea un cambio en la concepción de la estructura de la base de datos geográficos, mientras que el modelo de datos vectorial y raster estructuran sus datos en capas, este modelo intenta organizar la información en objetos geográficos y sus relaciones, estos objetos son agrupados en clases y son sometidos a una serie de procesos.

Introducen un carácter dinámico a la información incluida en el sistema, por ello es aconsejable usar este modelo en situaciones en que la naturaleza de los objetos que modelemos cambien constantemente en el tiempo y/o el espacio.

La ventaja fundamental que permite esta estructura de datos frente a las demás es la forma dinámica en que se representan los datos. Es decir, a partir de una serie de parámetros establecidos en el comportamiento de los objetos geográficos, podemos simular su evolución futura, lo que constituye un gran avance si se trabaja en entornos en los que se requiere simulación de situaciones potenciales.

1.4.1 Aplicaciones de los SIG.

Administrar, regular, controlar y planificar las acciones que se desarrollan en un territorio determinado constituye una tarea muy compleja. Por esto los SIG son utilizados en distintos sectores como una herramienta de apoyo y consulta. A continuación se brindan algunas aplicaciones de los de estos:

- Agricultura: Para el monitoreo y manejo de parcelas y granjas, estudios regionales y nacionales.
- Estudios Ambientales: Para evitar la degradación ambiental. Contribuye a determinar zonas de riesgo; ya sea por desertificación y deforestación. Para realizar estudios climatológicos.
- Epidemiología y Salud: Contribuye a la ubicación de enfermedades relacionadas con factores ambientales y actividades humanas.
- Turismo: Para la realización y manejo de facilidades y atracciones.
- Estudios sociales: Análisis de dinámicas demográficas.
- Centros de Gestión de Emergencias de Seguridad Ciudadana: Para visualizar en un área geográfica lo que se requiera atender; como la ubicación de eventos que son informados al centro por Organismos o Personas que requieran de su apoyo de los órganos de seguridad; la ubicación donde ha ocurrido un determinado incidente, la de sitios referenciados geográficamente como Estaciones de Bomberos, Policías, Hospitales y los recursos en servicio con Sistema de

Posicionamiento Global. Con los SIG no solo se logrará visualizar, los usuarios que interactúen con este podrán acceder y conocer los detalles de las estas informaciones que se mapifican y de los procedimientos generales que sean elaborados por parte del centro para enfrentar cualquier situación. Todo esto permitirá mantener informado a los Órganos de Seguridad.

1.5 Sistemas de Posicionamiento

Desde siempre el ser humano ha tratado de encontrar la forma de orientarse y saber el lugar exacto de su localización. Para resolver este problema se han buscado distintas vías de solución como orientarse por las constelaciones, por la posición del sol, etc. Hoy en día, gracias a la tecnología, ya no se trata de buscar estrellas que nos sitúen el Norte, sino de localizar satélites que están en órbita sobre la Tierra y nos indican nuestra posición exacta. Eso se consigue gracias a los receptores de los sistemas de navegación por satélite que permiten conocer con gran precisión dónde nos encontramos.

Los sistemas de posicionamiento constituyen uno de los avances tecnológicos más importante de las últimas décadas. Estos permiten que pequeños dispositivos electrónicos, denominados receptores determinen la localización (longitud, latitud y altitud) de cualquier persona u objeto con errores de algunos metros, usando señales de radio que transportan datos de tiempo, que son transmitidas en línea recta desde los satélites.

Existen varios modos para capturar la información e incorporarla a un SIG, como los GPS (Sistemas de Posicionamiento Global), GPS Diferencial (DGPS), GLONASS (GLObal NAVigation Satellite System), Galileo y Beidou.

1.5.1 Sistema de Posicionamiento Global (GPS).

Una de las maneras más utilizadas para capturar información geográfica e incorporarla a un SIG lo constituye el GPS. Este término procede del acrónimo de la expresión inglesa '**Global Positioning System**' (Sistema de Posicionamiento Global). Se trata de un sistema que permite calcular las coordenadas de cualquier punto de la superficie terrestre a partir de la recepción de señales emitidas desde una constelación de satélites en órbita. Básicamente, su principal funcionalidad es que permite al usuario conocer, mediante un receptor, su posición en cualquier parte del planeta [2].

El sistema GPS es un sistema compuesto por una red de 24 satélites denominada NAVSTAR, situados en una órbita a 20.000 km. de la Tierra, y unos receptores GPS, que permite determinar nuestra posición en cualquier lugar del planeta, de día o de noche y bajo cualquier condición meteorológica. Esta red de

satélites fue creada por el Departamento de Defensa de los Estados Unidos con fines militares para sustituir a un antiguo sistema: Doppler, sobre la constelación Transit.

Estos satélites llevan a bordo un reloj atómico de gran precisión y emiten señales personalizadas que indican la hora en que partieron. El receptor en tierra, reconoce el satélite que ha emitido la señal, determinando el tiempo que ha tardado en llegar y calculando la distancia que le separa de él. Una vez que el receptor GPS ha contactado con al menos cuatro satélites es capaz de determinar su longitud, latitud y altura.

Los receptores GPS tienen un error nominal en el cálculo de la posición de aprox. 15 m. que pueden aumentar hasta los 100 m. Para algún usuario que requiera un uso normal de este equipo esta diferencia no es muy significativa, pero para un Sistema de Gestión de Emergencias de Seguridad Ciudadana, este margen de error puede significar una catástrofe, por lo cual debe ser prácticamente nulo.

La utilización de GPS Diferencial (DGPS) es una solución a estos problemas de cálculo de posición pues reduce el margen de error a menos de un metro de diferencia con la posición indicada. Este tipo de receptor, además de recibir y procesar la información de los satélites, recibe y procesa, simultáneamente, otra información adicional procedente de una estación terrestre situada en un lugar cercano y reconocido por el receptor. Esta información complementaria permite corregir las inexactitudes que se puedan introducir en las señales que el receptor recibe de los satélites. En este caso, la estación terrestre transmite al receptor GPS los ajustes que son necesarios realizar en todo momento, éste los contrasta con su propia información y realiza las correcciones mostrando en su pantalla los datos correctos con una gran exactitud [3].

1.6 Metodología de desarrollo de software.

En la Industria de Software, hay tendencia al crecimiento del volumen y complejidad de los productos, se quiere un software que este mejor adaptado a nuestras necesidades y en el menor tiempo posible.

Sin embargo, la mayoría de los desarrolladores de hoy en día hacen sus sistemas con los mismos métodos de años atrás donde muchos proyectos fracasaban, sin percatarse que el elemento que hace que un software triunfe es tener un proceso bien definido y bien gestionado.

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software [4]

Durante el proceso de desarrollo de software se sigue una metodología que no es más que un conjunto de procedimientos, técnicas, herramientas y se genera una documentación, que indicará a los desarrolladores durante el desarrollo del software que es lo que hay que obtener en cada ciclo.

1.6.1 Extreme Programing (XP).

Es una de las metodologías de desarrollo más exitosas en la actualidad, utilizada en proyectos de corta duración y que tienen un pequeño equipo de desarrollo. Consiste en una programación rápida y extrema. Tiene una particularidad y es que el cliente forma parte del equipo de desarrollo.

XP se define especialmente para proyectos con requisitos imprecisos y muy cambiantes. A diferencia de otras metodologías no se genera ninguna documentación formal durante el proceso de desarrollo.

Se basa en tres características fundamentales:

- Pruebas Unitarias: son las pruebas realizadas a los principales procesos durante el desarrollo, lo que permite detectar las fallas en cualquier momento del desarrollo del software.
- Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible el cambio y permitiendo un continuo perfeccionamiento.
- Programación en pares: consiste en la participación de dos desarrolladores en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está desarrollando.

Esta metodología promueve la comunicación, entre los usuarios y los desarrolladores; la simplicidad, al desarrollar y codificar los módulos del sistema; y la retroalimentación, concreta y frecuente del equipo de desarrollo, del cliente y los usuarios finales.

1.6.2 Proceso Unificado de Desarrollo de Software (RUP).

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla, que explica los pasos necesarios para terminar el proyecto. [4]

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

En RUP se han agrupado las actividades en grupos lógicos, definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Divide en 4 fases el desarrollo del software:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- Transmisión: El objetivo es llegar a obtener el release del proyecto.

RUP es más adaptable para proyectos de largo plazo, con un equipo de desarrollo muy grande. En esta metodología se genera gran volumen de información, lo que posibilita un mayor entendimiento entre el equipo de desarrollo. Genera un gran número de artefactos y tiene abundantes roles.

Se basa en tres características esenciales que la distinguen de otras metodologías: Dirigido por Casos de Uso, Centrado en la Arquitectura, Iterativo e Incremental. Esto es lo que hace único al Proceso Unificado [4].

- Dirigido por Casos de Uso: Para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean [4]. Un caso de uso es un servicio que el actor requiere del sistema y este le proporciona un resultado. Representan los requerimientos funcionales de la aplicación. Los casos de uso guían el diseño, la implementación y prueba, es decir, guían el proceso de desarrollo.
- Centrado en la Arquitectura: La arquitectura en un sistema de software se describe mediante diferentes vistas del sistema en construcción. Esta se refleja en los casos de uso pues cada uno describe una funcionalidad y una forma. Es a su vez la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.
- Iterativo e Incremental: Es mucho más sencillo y práctico dividir el trabajo en pequeños módulos, los que no son más que iteraciones que resultan en un incremento. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software. Los beneficios de las iteraciones son los siguientes:
 - Reduce el coste del riesgo al coste de un solo incremento.
 - Menos riesgo de no sacar el producto al mercado en fecha.
 - Acelera el ritmo de desarrollo.
 - Las necesidades del usuario y correspondientes requisitos no pueden definirse completamente al principio. Se requieren iteraciones sucesivas.

RUP brinda un proceso integrado que utiliza el estándar de notación UML para permitir desarrollar un proceso de forma iterativa e incremental a partir de la identificación e implementación de los casos de uso.

1.6.3 Selección de la metodología adecuada.

Como hemos apreciado, a diferencia de XP, la metodología RUP es más adaptable para proyectos de largo plazo, con un equipo de desarrollo muy grande. En esta metodología se genera gran volumen de información, lo que posibilita un mayor entendimiento entre los miembros del equipo de desarrollo, respaldando de cierta forma la inestabilidad que se genera dentro del mismo. Otra de las diferencias es que se genera un gran número de artefactos que contribuyen a un mejor entendimiento del problema. Por todo lo antes expuesto se decidió que de las metodologías antes presentadas la que más se adapta a las características del proyecto es RUP.

1.7 Lenguaje de Modelado Unificado (UML).

Durante casi una década, el Lenguaje Unificado de Modelado (UML) ha sido el estándar de la industria para visualizar, especificar, construir y, documentar los artefactos de los sistemas software. Permite, además, conocer, configurar, mantener y controlar la información, pretendiendo unificar y estandarizar las técnicas de modelado.

Su objetivo inicial era simplificar y consolidar el gran número de métodos de desarrollo orientado a objetos que existían.

UML combina conceptos comúnmente aceptados por distintos métodos orientados a objetos, determinado definición, notación y terminología. Es representativo, involucra todo el ciclo de vida de desarrollo, está pensado para varios lenguajes y plataformas.

No es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación y plataformas de objetos distribuidos.

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

UML es un estándar, su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyecto. El resultado fundamental de UML es: ahorro de costos, mejor calidad de aplicaciones e importantes reducciones de tiempo en los proyectos de desarrollo.

1.8 Plataformas de Desarrollo.

1.8.1 Plataforma Microsoft .NET.

La plataforma Microsoft .NET proporciona las herramientas y tecnologías necesarias para desarrollar aplicaciones Web distribuidas. Expone un modelo de programación para aplicaciones multinivel, consistente e independiente del lenguaje de programación, a la vez que proporciona una interoperabilidad y una fácil migración desde las tecnologías actualmente existentes. Los lenguajes que permite utilizar esta plataforma comparten el mismo entorno de trabajo: el Framework .NET, que permite programar, compilar y ejecutar las aplicaciones.

Framework .NET

El Framework .NET proporciona los bloques básicos para desarrollar aplicaciones web y servicios web. A grandes rasgos, está formado por un runtime universal (Common Language Runtime, o CLR), una librería de clases unificada común (Base Class Library) y ASP.NET, con formularios web, servicios web y servicios de acceso a datos.

El Common Language Runtime (CLR) es un motor de ejecución de código de alto rendimiento. En particular, características como la gestión del tiempo de vida, nombres fuertemente tipados, gestión de excepciones multilenguaje, gestión de eventos basada en delegados, enlace dinámico o reflexión, reducen la cantidad de código que un desarrollador debe escribir para implementar la lógica de negocio de un componente reutilizable.

El Framework .NET también incluye un conjunto de librerías de clases (APIs) extensible, unificado, orientado a objetos y jerárquico que los desarrolladores pueden utilizar; la principal ventaja es que ya no necesitarán aprender múltiples librerías incompatibles de lenguajes independientes para realizar el trabajo. Entre estas librerías se encuentran las bibliotecas de clases: Microsoft ASP.NET para aplicaciones web y servicios web, Windows Forms para Aplicaciones Desktop y Microsoft ADO.NET para el acceso a datos.

Muchas características de .NET están diseñadas específicamente para mejorar la calidad de las herramientas que lo utilizan, como las funciones incorporadas de depuración y creación de perfiles.

Microsoft Visual Studio .NET, el entorno de desarrollo líder, está diseñado para aprovechar las ventajas que ofrece el Framework.

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web, servicios web, aplicaciones de escritorio y aplicaciones móviles.

Visual Studio .NET

Entre los lenguajes de programación que Visual Studio .NET permite desarrollar se encuentran: Visual C++, Visual C#, Visual Basic .NET y ASP.NET, entre otros. Seguidamente realizaremos una breve explicación de C# (o Visual C#).

Lenguaje de Programación C#

Microsoft Visual C# es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C. Ofrece un conjunto de construcciones que permiten facilitar la programación al desarrollador. Entre las particularidades que lo caracterizan se encuentran los siguientes:

- Orientado a Objetos.
- Modernidad.
- Distribuido.
- Seguridad de tipos.
- Instrucciones seguras.
- Sistema de tipos unificado.
- Extensibilidad de tipos básicos.
- Orientación a componentes.
- Eficiencia.
- Compatibilidad.

C# juega un importante papel en Visual Studio .NET porque ha sido diseñado para trabajar de forma óptima con este, de hecho, algunas bibliotecas de .NET como Collection, XML, ADO+, ASP+, GDI+ y otras fueron escritas en C#.

1.8.2 Plataforma Java:

La plataforma Java es el nombre de un entorno o plataforma de desarrollo creada por Sun Microsystems.

Se ejecuta sobre otra plataforma hardware/software y provee:

- El lenguaje de programación Java
- La Máquina Virtual Java (JVM)
- La Interfaz de Programación de Aplicaciones (API)

Esta plataforma ha evolucionado en concordancia con el avance tecnológico y se ha convertido en una de las plataformas de programación más usadas por los desarrolladores. Su principal ventaja es que su entorno de desarrollo es independiente de la plataforma sobre la que se trabaje, es decir, sus aplicaciones son funcionales tanto en Linux, Unix, Solaris, Power/Mac como en Windows, brindando una compatibilidad a nivel de fuentes, código intermedio y bibliotecas de clases.

La implementación en dicha plataforma proporciona un compilador de Java para traducir código fuente al conjunto de instrucciones de la máquina virtual de Java, un intérprete para ejecutar las instrucciones de la máquina virtual y una implementación de la API de Java.

Actualmente existen distintas ediciones o especificaciones de la plataforma Java como por ejemplo:

- J2ME (Java 2 Edición Micro): Orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant o Asistente Personal Digital), etc.
- J2EE (Java 2 Edición Empresarial): Orientada a entornos distribuidos empresariales o de Internet.
- J2SE (Java 2 Edición Estándar): Para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en una PC de escritorio.

Lenguaje de programación Java.

Java es un lenguaje de programación independiente de la plataforma, creado por Sun Microsystems. Alcanzó su madurez con la popularización de Internet y es en cierta manera el heredero legítimo de C++, eliminando la mayoría de sus complejidades como por ejemplo: la herencia múltiple y la creación de punteros.

Java presenta características que lo convierten en un lenguaje seguro, estándar y de alto nivel, algunas de las principales características se muestran a continuación:

- **Orientado a Objetos:** Basado en C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Soporta las tres características propias de la orientación a objetos: encapsulación, herencia y polimorfismo.

- **Distribuido:** Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. La característica de ser distribuido es debido a que proporciona las librerías y herramientas para que los programas puedan distribuirse, es decir, que se corran en varias máquinas, interactuando.
- **Interpretado:** Se traduce el código fuente a un código intermedio (bytecode), que es interpretado por La Máquina Virtual de Java, lo cual permite que se pueda ejecutar en cualquier sistema operativo.
- **Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.
- **Seguro:** No se permite el acceso ilegal a memoria ya que no se trabaja con punteros.
- **Portabilidad:** Una de las principales características de Java es su portabilidad. En lugar de compilarse a código nativo de la máquina, los programas de Java se traducen al formato bytecode que es el mismo en cualquier sistema operativo. Estos bytecodes son procesados directamente por La Máquina Virtual de Java que es dependiente de la máquina en uso. La portabilidad radica en que se pueden programar las aplicaciones solo una vez y ejecutar en cualquier plataforma. Por ejemplo, cuando se compila un programa Java en una plataforma Windows/Intel, se obtiene la misma salida compilada (o los mismos bytecodes) que en un sistema Macintosh o Unix
- **Altas prestaciones:** No se pierde tiempo optimizando código que no se ejecutará.
- **Multihilo:** Permite la ejecución de varias tareas a la vez.
- **Dinámico:** No conecta todos los módulos que comprende una aplicación hasta el tiempo de ejecución.

La mayoría de los lenguajes de programación se caracterizan por ser interpretados o compilados, lo que determina la manera en como serán ejecutados en una computadora. Java tiene la característica de ser al mismo tiempo compilado e interpretado. El compilador es el encargado de convertir el código fuente de un programa en bytecodes que son independientes de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java.

La programación en Java, permite el desarrollo de aplicaciones tanto bajo el esquema Cliente/Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar.

La Interfaz de Programación de Aplicaciones (API).

La API de Java está formada por una amplísima jerarquía de clases ya desarrolladas que ofrecen un gran abanico de posibilidades al programador, cubriendo una gran cantidad de aspectos relacionados con el desarrollo de software en general. La API es vastísima, está organizada en paquetes (packages) ordenadas por temas. La nomenclatura de los paquetes es uniforme y ayuda a categorizar las clases.

El J2SE permite la utilización de todos estos packages en el desarrollo de programas Java y el JRE (Java Runtime Environment) permite la ejecución de programas que usan cualquiera de las clases del API.

La Máquina Virtual Java (JVM).

Recibe este nombre porque es una máquina imaginaria que se implementa emulando por software una máquina real. Es un programa ejecutable para una plataforma específica, capaz de interpretar y ejecutar el bytecode, el cual es generado por el compilador del lenguaje Java. Su misión principal es la de garantizar la portabilidad de las aplicaciones Java.

Las tareas principales de la JVM son las siguientes:

- Reservar espacio en memoria para los objetos creados.
- Liberar la memoria no usada (garbage collector).
- Asignar variables a registros y pilas.
- Llamar al sistema huésped para ciertas funciones, como los accesos a los dispositivos.
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java.

En la JVM se encuentra el motor que en realidad ejecuta el programa Java y es la clave de muchas de las características principales de Java, como la portabilidad, la eficiencia y la seguridad.

1.8.3 Similitudes entre J2EE y .NET

J2EE es la especificación de la plataforma Java orientada a entornos distribuidos empresariales o de Internet. A continuación veremos algunas semejanzas entre J2EE y .NET.

- El propósito tanto de J2EE como de la plataforma .NET es facilitar y simplificar el desarrollo de aplicaciones empresariales o corporativas.
- Los servidores de aplicaciones J2EE y .Net proporcionan un modelo de acceso de componentes a datos y de lógica del negocio, separados por una capa intermedia de presentación implementada mediante ASP.Net (.Net) ó Servlets (J2EE).
- Visual Basic.Net y C# son lenguajes orientados a objetos, al igual que Java, y en su diseño ha tenido mucha importancia la existencia de Internet.

- Desde la perspectiva de los desarrolladores, J2EE y .Net proporcionan las herramientas necesarias para crear servicios web.

1.8.4 Ventajas de J2EE frente a .Net

A continuación veremos algunas ventajas de la plataforma J2EE frente a .Net:

- Las implementaciones de J2EE pueden adquirirse a distintas compañías, mientras que .Net solo puede comprarse a Microsoft.
- El hecho de que haya distintas organizaciones implementando J2EE ofrece mayor variedad para los usuarios y permite la existencia de una cierta competencia entre ellas para obtener mejores productos
- Las aplicaciones Java pueden correr en una amplia gama de sistemas operativos y de arquitecturas hardware. La plataforma .Net corre solamente sobre sistemas operativos de Microsoft siendo J2EE un entorno de desarrollo que ofrece una independencia real de la plataforma.
- La tecnología Java es una tecnología abierta y se basa en gran parte en estándares de organizaciones de normalización y estándares empresariales. Esto posibilita que los desarrolladores puedan conocer y entender la plataforma Java y aprovecharla para sus aplicaciones y, por otro lado, al basarse en estándares empresariales, simplifica la integración con productos de múltiples compañías. En contraposición, solo el código fuente del lenguaje C# de la plataforma .Net ha sido abierto al público general
- Aunque Java fue creado originalmente por la compañía Sun Microsystems, J2EE es ahora el producto de la colaboración de muchas empresas y organizaciones de todo tipo, es decir, es Open Source. La Plataforma .Net es el producto de una sola compañía: Microsoft.
- La tecnología Java goza de más madurez que .Net y ha sido probada su eficacia en muchos entornos y situaciones empresariales distintas.

1.8.5 Selección de la plataforma adecuada.

Como se ha apreciado, las plataformas Java y .Net tienen muchas características comunes que las hacen candidatas a la hora de construir aplicaciones empresariales distribuidas. No obstante, la plataforma Java a diferencia de .Net corre en múltiples sistemas operativos y múltiples máquinas hardware, cuyos usuarios pueden seleccionar la implementación que más les convenga. Java ha hecho realidad el sueño

de "escriba el código una vez, ejecútelo en cualquier parte", brindando una verdadera portabilidad de las aplicaciones construidas en dicha plataforma.

Por todo lo expuesto anteriormente, y teniendo en cuenta las características de la aplicación que estamos diseñado, se decidió usar la Plataforma Java con el lenguaje de programación Java y su especificación para aplicaciones empresariales: J2EE.

1.9 MapInfo.

MapInfo es líder de mercado en herramientas de software para la visualización y análisis de datos corporativos a través de mapas.

Los diferentes productos de MapInfo han sido diseñados para que los usuarios visualicen y examinen los datos desde una perspectiva geográfica, superponiéndolos en mapas digitales con diferentes niveles. De esta forma, es posible utilizar la información geográfica de una empresa (por ejemplo, ciudades, nombres de calles, códigos postales, etc.) para descubrir modelos de comportamiento y tendencias que resultan difíciles de identificar de otra manera.

MapInfo ofrece una amplia gama de soluciones y software para PC e Internet, herramientas de desarrollo de aplicaciones y datos.

Los archivos de MapInfo se estructuran alrededor de las denominadas tablas que contienen la información espacial y la de atributos. De esta forma, por una tabla lo mismo podemos entender un mapa o una capa. La información se almacena en tablas independientes que generalmente conforman 4 ó 5 ficheros enlazados entre si, los cuales se crean automáticamente y mantienen el mismo nombre, estos serán:

- Nombre de la **tabla.TAB** (fichero de textos con el encabezamiento de toda la información).
- Nombre de la **tabla.DAT** (fichero binario y de textos con los datos alfanuméricos).
- Nombre de la **tabla.MAP** (fichero binario con los datos espaciales)
- Nombre de la **tabla.ID** (fichero binario para el enlace de los demás ficheros).
- Nombre de la **tabla.IND** (fichero binario con los campos de la base de datos que han sido indexados) (opcional).

A continuación se mencionan algunos de los principales productos de MapInfo que de una forma u otra facilitan el trabajo a los usuarios de los SIG.

- **MapInfo MapBasic:** Entorno de desarrollo de aplicaciones para MapInfo Professional.

- **MapInfo MapX:** La forma más fácil y económica de añadir funcionalidad de gestión de mapas a otras aplicaciones y soluciones.
- **MapInfo MapXtreme 2005:** Es la novedad principal del entorno de desarrollo basado en ubicaciones. Permite crear aplicaciones personalizadas sobre la plataforma .NET
- **MapInfo SpatialWare:** Herramienta de gestión de información para almacenar, gestionar y manipular los datos basados en ubicación.
- **Mapinfo MapXtreme Java Edition:** Potente servidor de mapas para Internet basado en Java, para implementaciones globales de las aplicaciones de mapas.

1.9.1 Mapinfo MapXtreme Java Edition,

Es un servidor de gestión de mapas para la visualización geográfica y la toma de decisiones, que permite el desarrollo y la implantación de aplicaciones con rapidez y bajo coste. Ofrece las siguientes funcionalidades:

- Es compatible con los principales servidores web y de aplicaciones.
- Compatible con MapInfo SpatialWare (producto de almacenamiento de datos espaciales).
- No utiliza plug-in, de manera que el cliente puede utilizar cualquier navegador web para hacer peticiones al servidor de mapas.
- Brinda un Framework completo, aplicaciones de ejemplo y una gran documentación diseñada para acelerar el desarrollo de aplicaciones.
- Fácil de personalizar, los JavaBeans pueden ser usados en entornos de desarrollo visual.
- Permite la conexión a Bases de Datos Oracle Spatial 10g.
- Es multiplataforma.

Para la gestión de los mapas propone:

- Mapas temáticos (permite sombreado, gráficos de tarta y de barra).
- Gestión de objetos (permite guardar, intersectar o borrar objetos como puntos, líneas y polígonos, etc.).
- Capa modificable (permite dibujar objetos personalizados).
- Búsqueda (permite localizar recursos rápidamente en un mapa, ejemplo: calle, código postal, etc.).
- Control de capas (permite gestionar múltiples capas, como colores de capas de datos, niveles de zoom, visibilidad y estilos de etiquetas).

- Selecciones espaciales (permite trabajar con datos espaciales de una región específica, ejemplo una provincia).

1.10 Herramientas CASE.

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo en gran medida el costo de la misma en tiempo y dinero.

Estas herramientas permiten poner en práctica las metodologías, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo, facilitando la realización de prototipos y el desarrollo conjunto de aplicaciones; mejorando y estandarizando la documentación que se genera en el proceso de desarrollo de software. Además aumentan la portabilidad de las aplicaciones; facilitan la reutilización de componentes de software y permiten desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Sin lugar a dudas las herramientas CASE han venido a revolucionar la forma de automatizar los aspectos claves en el desarrollo de los sistemas de información, debido a la gran plataforma de seguridad que ofrecen a los sistemas que las usan. Desde que se crearon éstas herramientas (1984) hasta la actualidad, cuentan con una credibilidad y exactitud que gozan de un reconocimiento universal.

Existen numerosas herramientas CASE para el modelado de software, pero en este trabajo se realiza un estudio de Microsoft Office Visio 2003, del Visual Paradigm y el Rational Rose Enterprise Edition.

1.10.1 Microsoft Office Visio 2003.

Microsoft Office Visio 2003 es una solución para la creación de gráficos y diagramas que ayudan a transformar, documentar, diseñar procesos y sistemas complejos de una forma rápida y clara.

Presenta un entorno amigable y flexible, que brinda facilidades a la hora de exportar diagramas. Brinda la funcionalidad de Guardar como página Web.

Microsoft Office Visio 2003 es de gran utilidad ya que brinda componentes que facilitan la elaboración de diagramas orientados al desarrollo de software, como el diseño de la interfaz de usuario para cada Caso de Uso.

1.10.2 Rational Rose Enterprise Edition.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que emplea durante el ciclo de vida del software la notación UML como lenguaje de modelado. Constituye un soporte al RUP.

Dentro de la suite del Rational Rose se encuentra Rational Rose Enterprise Edition que facilita la modelación de los procesos del negocio, captura de requisitos, análisis y diseño orientado a objetos, implementación del sistema mediante componentes y despliegue organizado en las diferentes vistas, que son: vista de casos de uso, vista lógica, vista de componentes y vista de despliegue.

Es una herramienta compatible solamente con sistemas operativos de Microsoft que permite generar documentación y código fuente a partir de un diseño UML, para lenguajes como son: Java, C++, Ada, Visual Basic, etc. Además admite ingeniería inversa, ya que a partir del código del programa se puede obtener información sobre su diseño.

Usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación. Acelera la implementación de sistemas con la calidad requerida.

1.10.3 Visual Paradigm.

Visual Paradigm es una poderosa herramienta de modelado visual que utiliza el lenguaje UML (Unified Modeling Language) para modelar los artefactos de un proyecto de software. Acelera el desarrollo del software completo obteniendo productos de calidad reduciendo costos y riesgos.

Posee entre sus principales características las siguientes:

- Es profesional: brinda la posibilidad de crear un conjunto bastante amplio de artefactos utilizados con mucha frecuencia durante el desarrollo de un Software.
- Es amigable: puede ser utilizado en varios idiomas, sus componentes se encuentran relacionados, por lo que se hace muy fácil la creación de cualquier tipo de diagrama, ya que cada componente utilizado en el diagrama que se esté creando, sugiere nuevos posibles componentes a utilizar.
- Brinda un número considerable de estereotipos a utilizar, lo que permite un mayor entendimiento de los diagramas.
- Facilidades para redactar especificaciones de casos de uso: es posible crear plantillas para las especificaciones de casos de uso y describirlos, por lo que no se necesita de una herramienta externa como editor de texto.
- Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código.
- Integración con distintos Entornos de Desarrollo Integrados (IDEs): se integra fácilmente con varios IDEs, entre ellos el de Visual Studio y el Eclipse.

- Interoperabilidad con otras aplicaciones: brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose. Además permite importar y exportar XML.
- Generación de código ORM: permite generar a partir de un Diagrama de Entidad Relación una Base de Datos Relacional y el código necesario para acceder a esta base de datos utilizando Java, PHP, C# o Enterprise Object Framework.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris.

1.10.4 Selección de las herramientas adecuadas.

A partir de los estudios realizados los cuales se describen en este epígrafe se decidió como herramientas CASE a Visual Paradigm ya que a diferencia de Rational Rose, es una herramienta multiplataforma, amigable y fácil de usar, brinda facilidades para generar documentación sin utilizar herramientas externas. Permite intercambiar información con Microsoft Visio, Rational Rose y XML. Se integra a IDEs como el Eclipse. Es muy fácil de usar en la creación de todo tipo de diagramas UML, para los que dispone de un número considerable de estereotipos que permiten un mayor entendimiento de los mismos.

Otras de las herramientas seleccionadas fue Microsoft Office Visio 2003 por las facilidades que brinda para crear diagramas orientados al desarrollo de software.

1.11 Macromedia Dreamweaver.

Macromedia Dreamweaver es un potente sistema profesional para crear, diseñar, desarrollar y mantener sitios web complejos y bien estructurados. Desde su lanzamiento en 1997 ha ido escalando en esta industria convirtiéndose en un estándar para el desarrollo web.

En el 2002 Macromedia lanzó Dreamweaver MX la cual permitió crear sitios web con codificación manual, diseño visual o la combinación de ambas. En septiembre de 2003 se lanzó Dreamweaver MX 2004 cuya ventaja fundamental sobre otras versiones era la incorporación de tecnología Cascade Style Sheet (CSS). Dreamweaver 8 es creado a partir de las versiones anteriores brindando un entorno de autoría avanzado con CSS, un mayor soporte para la integración de XML. Entre sus nuevas funcionalidades se encuentran:

- Auditoria visual con XML: Acelera el ritmo de trabajo con XML, siendo menos complejo transformar el XML a HTML.

- Nuevo panel CSS unificado: Es una herramienta integral para aprender, entender y trabajar de forma visual con los estilos CSS aplicados a las páginas. Permite ver mejor la lista de estilos aplicados a un determinado elemento.
- Visualización de diseños CSS: Define los bordes de los diseños en formato CSS o aplicar color a los diseños CSS visibles. Contiene ayudas contextuales de gran utilidad.
- Barra de herramientas de reproducción de estilos: Permite un acceso rápido a las vistas de diseño observando el resultado final que se obtendrá.
- Mejora en la reproducción de diseños CSS: Permite escoger la forma que se reproducen los diseños en formatos CSS en los navegadores.

Dreamweaver 8 es compatible con las principales tecnologías de servidor como por ejemplo: PHP, ASP, ASP.NET y JSP. [6]

1.12 Conclusiones.

Al realizar el estudio de los fundamentos teóricos que sustentan esta investigación, se constató la influencia de los SIG sobre web en los Centros de Gestión de Emergencias de Seguridad Ciudadana (171), especificando la importancia que tiene en la toma de decisiones en base a mejorar el servicio de las órganos encargadas de la seguridad ciudadana.

Además, se realizó un análisis detallado de la metodología a utilizar, de las herramientas CASE, de las plataformas de desarrollo sobre las cuales se podía desarrollar un software con las características de un SIG.

Después de un extenso proceso de investigación, se decidió que el SIG sobre web para este centro se analizará y diseñará utilizando: la Plataforma de Desarrollo Java, lenguaje de programación Java, metodología RUP y UML como lenguaje de modelación. Las herramientas CASE utilizadas serán Visual Paradigm y Microsoft Office Visio 2003, la herramienta para el diseño seleccionada es Dreamweaver 8 y MapInfo MapXtreme Java Edition como framework para mapas digitales.

Capítulo 2: Descripción de la Solución Propuesta.

2.1 Introducción

En este capítulo se presenta la propuesta de un SIG sobre web como apoyo a los Centros de Gestión de Emergencias de Seguridad Ciudadana (171) en la República Bolivariana de Venezuela que permitirá mantener informado a los Órganos de Seguridad Ciudadana

Se muestra un modelo de dominio como alternativa al negocio, modelando y relacionando los principales conceptos que se identificaron en el campo de acción. Además se enumeran los requerimientos funcionales del sistema que se propone. Se identificaron los actores y Casos de Uso (CU) del sistema, obteniendo el Modelo de CU del Sistema empleando a RUP como metodología de desarrollo y UML como lenguaje de modelado.

2.2 Modelo de Dominio

Teniendo en cuenta que en el Centro de Emergencias de Seguridad Ciudadana (171) actualmente no existe experiencia de trabajo con un SIG sobre web, por lo que los procesos en este sentido no están claramente definidos, se propone realizar un Modelo de Dominio, donde se relacionan los principales conceptos identificados en el campo de acción y que se utilizarán en el desarrollo del SIG sobre web.

Esto ayudará a que los usuarios, desarrolladores e interesados en la aplicación puedan utilizar un vocabulario común en aras de lograr una mayor comprensión del problema, y que contribuya a una correcta captura de requisitos y a una construcción exitosa del sistema.

2.2.1 Conceptos del Modelo del Dominio:

- **Usuario:** Son los actores que interactúan con el mapa.
- **Sistema de Información Geográfica (SIG):** Es un conjunto de capas superpuestas que es capaz de integrar, almacenar, editar, analizar y mostrar en pantalla información referenciada geográficamente.
- **Capa:** Es una colección de elementos referenciados geográficamente de un mismo tipo espacial y con los mismos atributos.
- **Elemento Georreferenciado:** Es la unidad de información básica que se representa en una capa. Los tipos espaciales en los que podemos encontrar los elementos son puntos, líneas o polígonos.
- **Punto:** Se traduce en coordenadas (latitud, longitud) o (x, y) del SIG donde se mostrará la imagen que representa a un determinado recurso, evento o solicitud.

- **Línea:** Es una serie ordenada de posiciones unidas por segmentos rectos. Permite modelar calles, ríos y los propios eventos.
- **Polígono:** Es un conjunto de líneas cerradas que delimitan superficies. Son empleados para modelar parroquias, municipios, cuadras.
- **Solicitud:** Constituye un incidente formulado por parte de la población al Centro de Gestión de Emergencia de Seguridad Ciudadana (171), que necesita respuesta de los Órganos de Seguridad Ciudadana.
- **Evento:** Constituye un suceso que ocurrirá en una fecha y hora determinada que es solicitado por un organismo o por una persona específica que necesita un apoyo de los Órganos de Seguridad Ciudadana.
- **Recursos:** Los recursos que se mapifican en un punto determinado son los siguientes:
 - Móvil: Se incluyen todos los vehículos ya sean o no propios del centro, como son: Motos, ambulancias, carros de bomberos, patrullas, etc.
 - Fijos: Pueden ser edificios, hospitales, escuelas, museos, cine, bancos, centros comerciales, teatros, iglesias, etc. Áreas abiertas que incluye los parques, plazas y bulevares. Referenciados donde se recogen aquellos que solo necesitamos la dirección como postes, cámaras de video de vigilancia, cabinas telefónicas, etc.

Los conceptos que anteriormente se precisaron son los que se relacionan entre si conformando la naturaleza del problema a analizar y diseñar. A continuación se muestra el Modelo de Dominio que se definió:

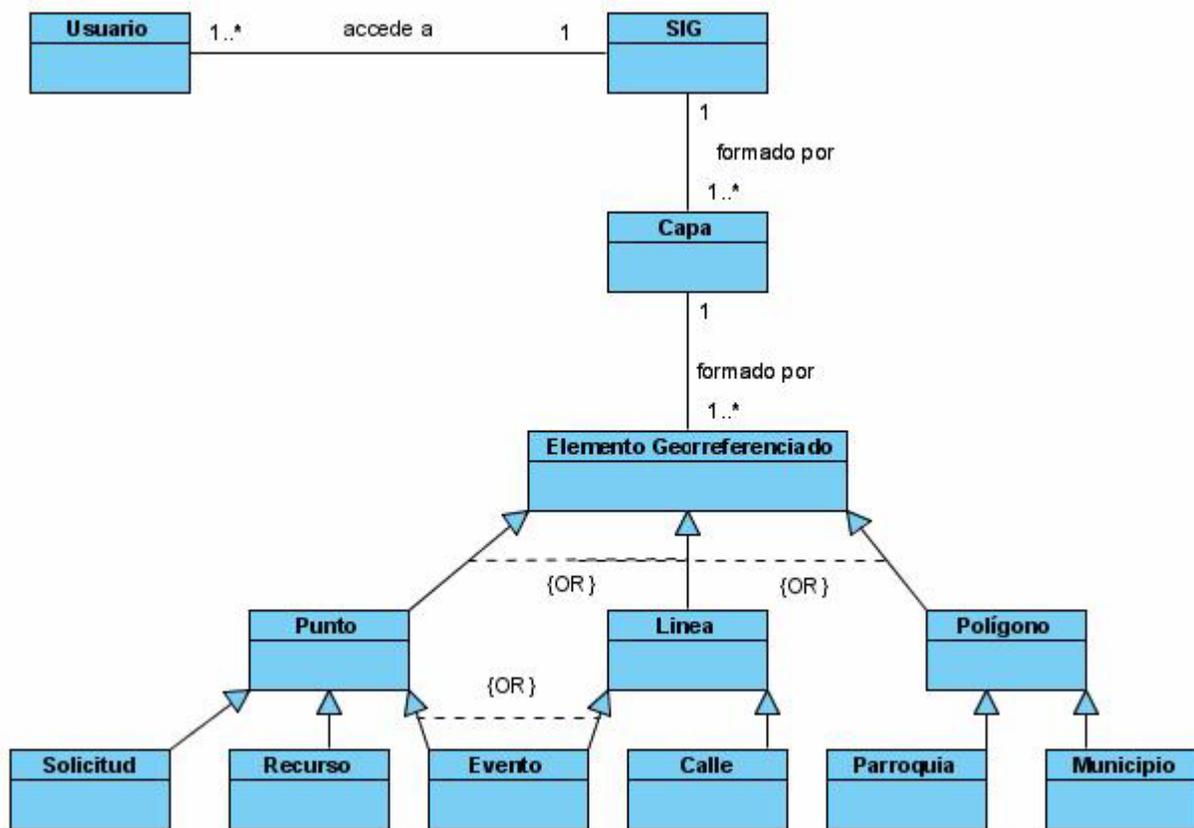


Fig 2.1. Modelo de Dominio.

2.2.2 Descripción del Modelo de Dominio.

Los usuarios que interactúan con el SIG podrán acceder a todas las funcionalidades que este le brinda.

Un SIG está formado por varias capas y cada una de esta por un conjunto de elementos de una misma geometría, es decir, un mapa está compuesto por varias figuras de diferentes tipos, líneas, polígonos y puntos, separadas en capas independientes.

Con el SIG se podrán realizar operaciones básicas como localizar realizar zoom, centrar mapa, abrir capa, manipular capa, etc. Además sobre este se muestran las calles, edificaciones, sitios de interés, solicitudes, recursos, eventos y procedimientos generales. Estas funcionalidades garantizarán mantener informado las 24 horas del día a los Órganos de Seguridad Ciudadana.

2.3 Requerimientos Funcionales.

A continuación se relacionan los requerimientos funcionales, estos son las condiciones o capacidades que el sistema debe cumplir.

R1: Autenticar Usuario.

1.1: Información a solicitar:

1.1.1: Usuario.

1.1.2: Contraseña.

1.2: Verificar que el usuario está registrado y que la contraseña es correcta.

1.3: Verificar que el usuario no está autenticado en otra sesión.

1.4: Almacenar fecha y hora en que inició la sesión y código del usuario.

R2: Cargar Configuración.

2.1: Mostrar la vista del mapa asociada al área de trabajo.

2.2: Habilita las opciones para el usuario autenticado.

R3: Acercar el área visible.

3.1: Ver el área visible del mapa con mayor grado de detalle.

R4: Alejar el área visible.

4.1: Ver el área visible del mapa con menor grado de detalle.

R5: Ver Vista Completa de Capa.

5.1: Ver vista completa de una o todas las capas.

5.1.1: Código de la capa.

R6: Mostrar Leyenda.

6.1: Muestra una lista de los iconos correspondientes a cada elemento mostrado en el mapa con su descripción.

6.1.1: Descripción.

6.1.2: Icono del elemento.

R7: Abrir Capa.

7.1: Abrir la capa seleccionada, añadiéndola a la lista de las capas abiertas:

7.1.1: Código de la capa.

R8: Cerrar Capa.

8.1: Cerrar la capa seleccionada, eliminándola de la lista de capas abiertas.

8.1.1: Código de la capa.

R9: Centrar Mapa.

9.1: Cambiar el área visible del mapa, tomando como nuevo centro el punto seleccionado.

9.1.1: Coordenadas del centro.

R10: Seleccionar región.

10.1: Selección regional:

10.1.1: Coordenadas del punto.

10.2: Selección rectangular:

10.2.1: Coordenadas de los vértices.

R11: Mostrar/Ocultar Capa en el mapa.

11.1: Permite Mostrar/Ocultar una capa sobre el mapa:

11.1.1: Código de la capa.

R12 Seleccionar/No Seleccionar Elemento.

12.1: Permite Seleccionar/No Seleccionar elementos sobre una capa abierta en el mapa:

12.1.1: Código de la capa.

R13 Mostrar/No Mostar Datos de la Capa.

13.1: Permite Mostrar/No Mostrar datos de una capa abierta en el mapa.

13.1.1: Código de la capa.

R14: Subir/Bajar Nivel.

14.1: Permite Subir/Bajar nivel de una capa abierta en el mapa:

14.1.1: Código de la capa.

14.1.2: Posición de la capa en mapa.

R15: Identificar Elemento.

15.1: Obtener del elemento seleccionado toda la información que se maneje del mismo en las capas mostradas en el mapa:

15.1.1: Código del elemento.

R16: Localizar Elemento.

16.1: Permite localizar un elemento en el mapa y centrar el mapa a partir de las coordenadas del mismo:

16.1.1: Código del elemento.

16.1.2: Coordenadas del elemento.

R17: Mostrar Elemento.

17.1: Buscar y mostrar los elementos que cumplen con el criterio de búsqueda introducido. El criterio de búsqueda es la combinación de estos dos campos:

17.1.1: Capa donde se encuentra el elemento.

17.1.2: Nombre del elemento.

17.2: Resultados de la búsqueda:

17.2.1: Nombre del elemento.

17.2.2: Código del elemento.

17.2.3: Coordenadas del elemento.

R18: Insertar Elemento.

18.1: Insertar un elemento en una posición determinada del mapa.

R19: Indicar los Recursos que tiene asignado una Solicitud.

19.1: Resalta en el mapa los recursos que tiene asignado una solicitud determinada:

19.1.1: Código de la solicitud.

R20: Mostrar detalles de una Solicitud.

20.1: Buscar y mostrar las solicitudes que cumplen con el criterio de búsqueda introducido. El criterio de búsqueda es la combinación de los siguientes datos:

20.1.1: Categoría de motivo.

20.1.2: Motivo.

20.2: Resultados de la búsqueda:

20.2.1: Código.

20.2.2: Motivo.

20.2.3: Estado.

20.2.4: Dirección.

20.3: Los detalles que se obtienen de la solicitud seleccionada son los siguientes:

20.3.1: Motivo.

20.3.2: Estado.

20.3.3: Dirección.

20.3.4: Punto de referencia.

20.3.5: Fecha.

20.3.6: Descripción.

R21: Mostrar detalles de un Evento informado por Organismo.

21.1: Buscar y mostrar los eventos informados por organismos que cumplen con el criterio de búsqueda introducido. El criterio de búsqueda es la combinación de los siguientes datos:

21.1.1: Fecha de ocurrencia del evento.

21.1.2: Categoría del evento.

21.2: Resultados de la búsqueda:

21.2.1: Código.

21.2.2: Estado.

21.2.3: Categoría.

21.3: Los detalles que se obtienen del evento seleccionado son los siguientes:

21.3.1: Detalles generales:

21.3.1.1: Fecha de ocurrencia.

21.3.1.2: Hora de ocurrencia.

21.3.1.3: Estado.

21.3.1.4: Categoría.

21.3.1.5: Dirección.

21.3.1.6: Detalles de la dirección.

21.3.1.7: Punto de referencia involucrado en la dirección del evento.

21.3.2: Detalles del organismo:

21.3.2.1: Nombre del organismo.

21.3.2.2: Cargo que ocupa la persona que solicitó el evento.

21.3.2.3: Nombre y apellidos de la persona que solicitó el evento.

21.3.2.4: Cédula de identidad de la persona que solicitó el evento.

21.3.2.5: Teléfono para contacto.

R22: Mostrar detalles de un Evento informado por Persona.

22.1: Buscar y mostrar los eventos informados por personas que cumplen con el criterio de búsqueda introducido. El criterio de búsqueda es la combinación de los siguientes datos:

22.1.1: Fecha de ocurrencia del evento.

22.1.2: Categoría del evento.

22.2: Resultados de la búsqueda:

22.2.1: Código.

22.2.2: Estado.

22.2.3: Categoría.

22.3: Los detalles que se obtienen del evento seleccionado son los siguientes:

22.3.1: Detalles generales:

22.3.1.1: Fecha de ocurrencia.

22.3.1.2: Hora de ocurrencia.

22.3.1.3: Estado.

22.3.1.4: Categoría.

22.3.1.5: Dirección.

22.3.1.6: Detalle de la dirección.

22.3.1.7: Punto de referencia involucrado en la dirección del evento.

22.3.2: Detalles del solicitante:

22.3.2.1: Nombre y apellidos de la persona que solicitó el evento.

22.3.2.2: Cédula de identidad de la persona que solicitó el evento.

22.3.2.3: Teléfono para contacto.

R23: Indicar la Solicitud a la que ha sido asignado un Recurso.

23.1: Resalta en el mapa la solicitud a la que ha sido asignado un recurso:

23.1.1: Tipo del recurso

23.1.2: Código de recurso.

R24: Mostrar Recurso.

24.1: Buscar y mostrar los recursos que cumplen con el criterio de búsqueda introducido. El criterio de búsqueda es la combinación de los siguientes datos:

24.1.1: Nombre del recurso.

24.1.2: Categoría del recurso.

24.1.3: Tipo del recurso.

24.2: Resultados de búsqueda:

24.2.1: Nombre del recurso.

24.2.2: Categoría del recurso.

24.2.3: Tipo del recurso.

24.3: Detalles del recurso seleccionado:

24.3.1: Fotos del recurso.

24.3.2: Lista de atributos.

24.3.2.1: Atributo.

24.3.2.2: Valor.

R25: Ver Detalles de un Recurso.

25.1: Ver los detalles de los recursos que coincidan con el criterio de búsqueda:

25.1.1: Código del recurso.

25.2: Resultados de la búsqueda:

25.2.1: Nombre del recurso.

25.2.2: Categoría del recurso.

25.2.3: Tipo del recurso.

25.2.4: Fotos del recurso.

25.2.5: Otros detalles:

25.2.5.1: Lista de atributos.

25.2.5.1.1: Atributo.

25.2.5.1.2: Valor.

R26: Actualizar posición de recursos.

26.1: Actualizar la posición de todos los recursos mapificados en el mapa:

26.1.1: Coordenadas actuales de cada recurso.

R27: Actualizar estado de recursos.

27.1 Actualizar el estado de todos los recursos mapificados en el mapa:

27.1.1: Estado actual de los recursos.

R28: Encuestar Velocidad de un Recurso.

28.1: Permite conocer la velocidad de un recurso seleccionado:

28.1.1: Código de un recurso móvil.

R29: Mostrar Procedimiento General.

29.1: Buscar y mostrar los procedimientos que cumplen con el criterio de búsqueda introducido. El criterio de búsqueda es:

29.1.1: Nombre del procedimiento general.

29.2: Resultados de la búsqueda:

29.2.1: Código del procedimiento general.

29.2.2: Nombre del procedimiento general.

29.2.3: Si tiene vista asociada al procedimiento general.

29.3: Detalles del procedimiento general seleccionado:

29.3.1: Descripción del procedimiento general.

29.3.2: De los recursos asignados:

29.3.2.1: Nombre del recurso.

29.3.2.2: Dirección donde está ubicado.

R30: Abrir Vista de Procedimiento General.

30.1: Se muestra la vista de un procedimiento general seleccionado en el mapa.

30.1.1: Código del procedimiento general.

R31: Ver información de la región seleccionada.

31.1: Obtener de la(s) región(es) seleccionada(s) toda la información que se maneje de la misma en las capas mostradas en el mapa:

31.1.1: Código de la(s) región(es).

2.4 Modelo de Caso de Uso (CU).

Para desarrollar el modelo de CU es necesario determinar los actores, los CU y las relaciones entre estos.

En este sistema solo interactúa un actor, el mismo se define a continuación:

Actores	Justificación
Usuario	Persona autenticada en el sistema que interactúa con él para consultar información de recursos, eventos y solicitudes. Puede abrir Procedimientos Generales y podrá realizar todas las operaciones básicas que se definan en el SIG.

Tabla 2.1. Actores del Sistema.

Para una mejor organización y comprensión del sistema que se propone se agruparon los CU en paquetes teniendo en cuenta las funcionalidades de los mismos.

A continuación se muestra el Diagrama de Paquetes del módulo:

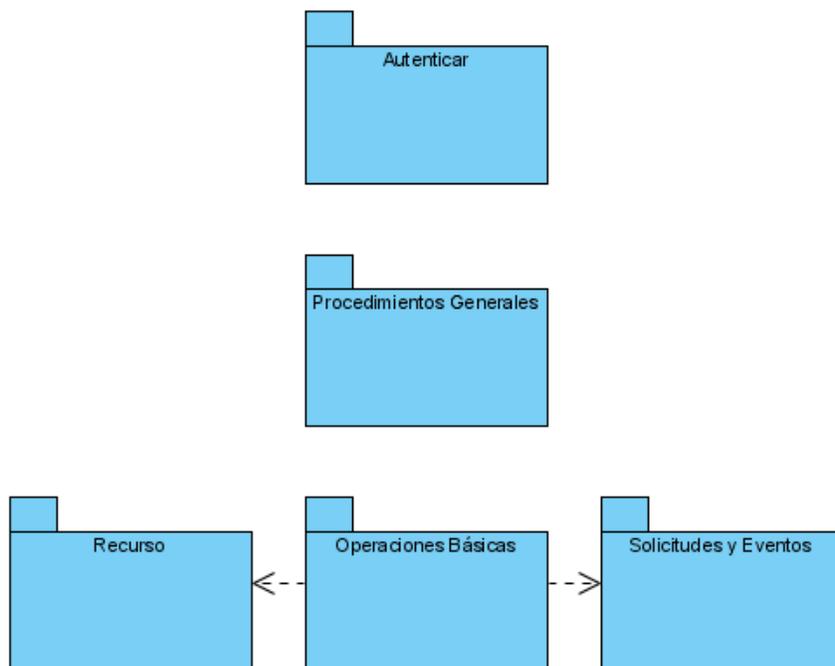


Fig 2.2 Diagrama de Paquetes. Módulo Mapificación Web.

A continuación se muestran los diagramas de CU por paquetes. Los CU que se representan por el color verde significan que pertenecen a otro módulo, y que se reutilizan aquí, se representan para obtener una mayor claridad del sistema.

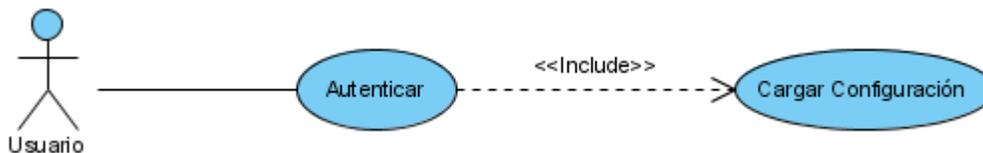


Fig 2.3 Diagrama de CU. Paquete Autenticar.

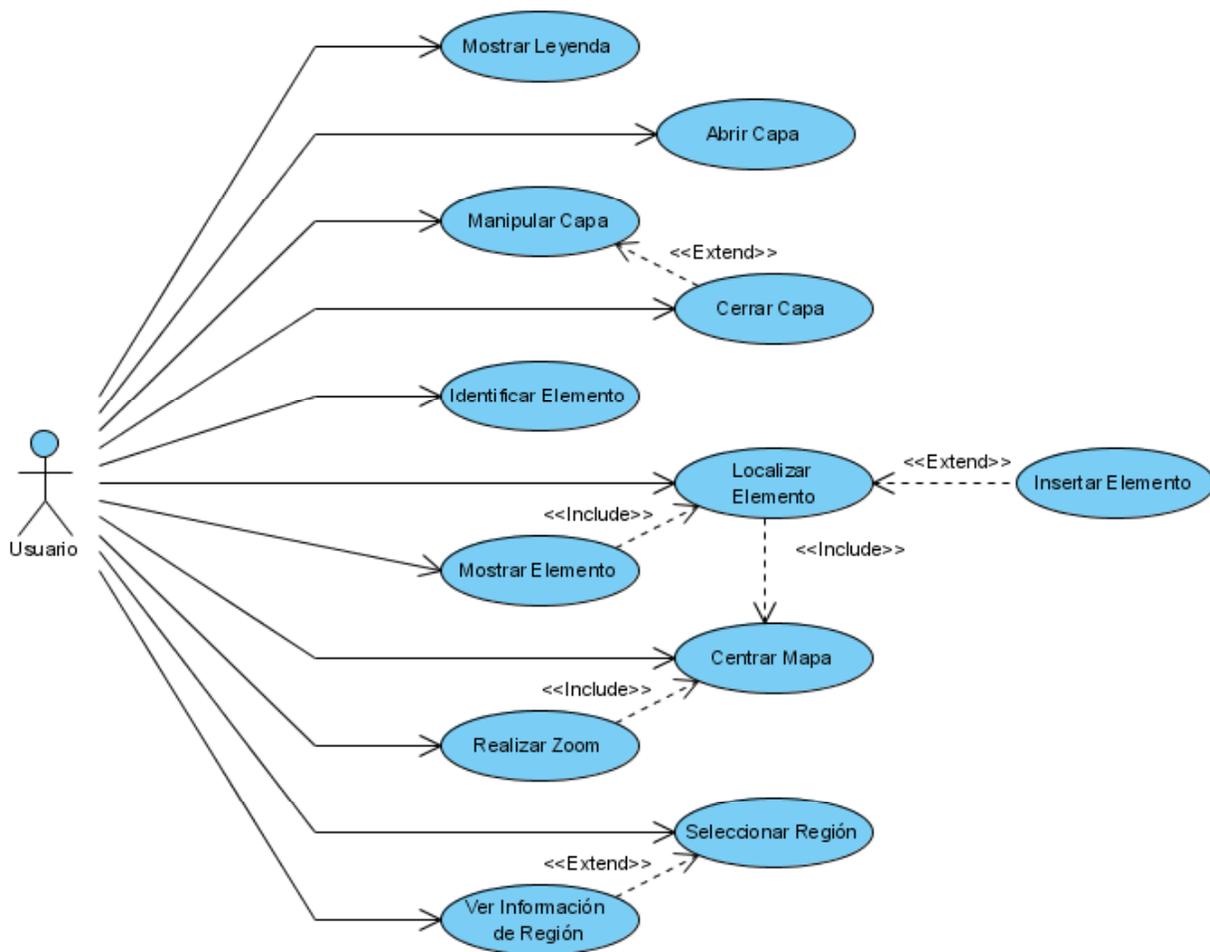


Fig 2.4 Diagrama de CU. Paquete de Operaciones Básicas.

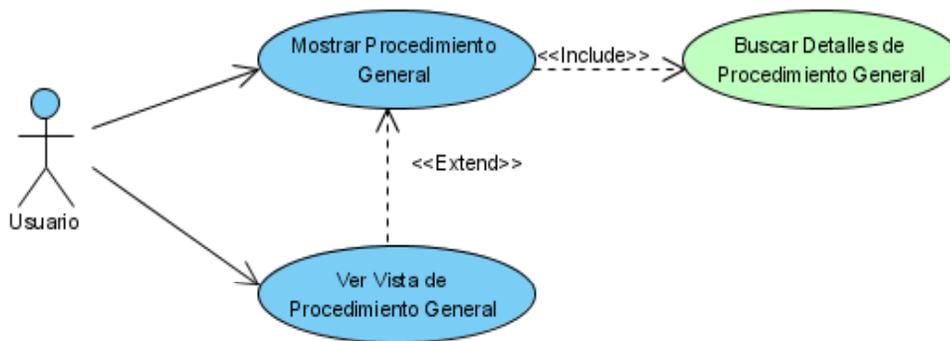


Fig 2.5 Diagrama de CU. Paquete Procedimientos Generales.

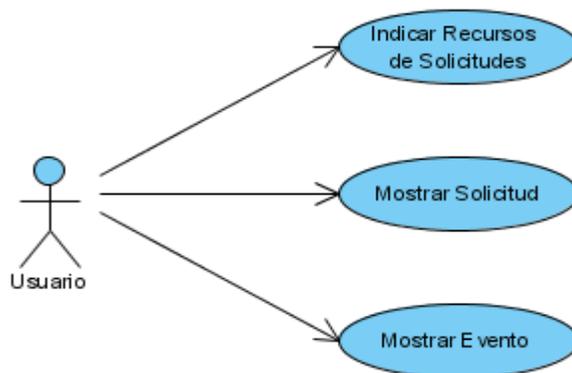


Fig. 2.6 Diagrama de CU. Paquete Solicitudes y Eventos.

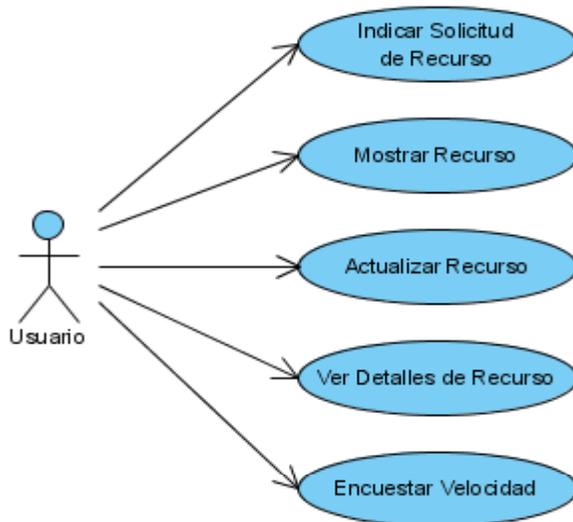


Fig. 2.7 Diagrama de CU. Paquete Recursos.

Todos los CU de este módulo se desarrollarán en el primer ciclo de desarrollo ya que conforman las funcionalidades básicas para trabajar con el sistema y las operaciones principales que pueden realizarse sobre los elementos referenciados geográficamente.

2.5 Descripción de los Casos de Uso.

Mediante la descripción de los CU se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. A continuación se presentan las descripciones de los casos de uso:

CU Autenticar Usuario

Propósito

Permitir que el usuario inicie sesión en la aplicación.

Descripción

El usuario introduce los datos para autenticarse y se verifica que los datos introducidos son correctos y que el usuario cuenta con los permisos necesarios para autenticarse en el sistema. En caso contrario no se permite la autenticación del usuario. También se determina si el usuario está autenticado desde otra PC.

Diseño de Interfaz de Usuario

Ver anexo 1 y 2.

Precondiciones

No aplica.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario introduce el nombre de usuario y la contraseña.	
2. El Usuario selecciona la opción "Aceptar"	3. Comprueba que exista conexión con la Base de Datos (BD).
	4. Comprueba que los campos no estén vacíos.

	5. Comprueba en la BD que el usuario se corresponde con un usuario del sistema y verifica que la contraseña pertenezca a este.
	6. Obtiene de la BD los siguientes datos del usuario autenticado: <ul style="list-style-type: none"> • Código del usuario. • Código del área a la que pertenece.
	7. Comprueba en la BD que no exista otra sesión abierta para ese usuario en la aplicación usando el nombre de usuario y la contraseña introducidos.
	8. Almacena en la BD la fecha y la hora en la que inició sesión y el código del usuario autenticado.
	9. Invoca al CU Cargar Configuración.
Flujo Alterno 3a No Existe Conexión con la BD	
Acción del Actor	Respuesta del Sistema
	3a.1. Comprueba que no existe conexión con la BD.
	3a.2. Muestra un mensaje indicando que no existe conexión con la BD.
3a.3 El usuario selecciona la opción "Aceptar".	3a.4 Cierra la aplicación.
Flujo Alterno 4a Existen Campos Vacíos	
Acción del Actor	Respuesta del Sistema

	4a.1. Comprueba que los campos usuario y/o contraseña están vacíos.
	4a.2. Indica los campos que están vacíos.
Flujo Alternativo 5a Datos del Usuario Incorrectos	
Acción del Actor	Respuesta del Sistema
	5a.1. Comprueba en la BD que los datos de autenticación no son correctos.
	5a.2 Muestra un mensaje indicando que el nombre de usuario y/o la contraseña proporcionados no son correctos.
Flujo Alternativo 7a Usuario Autenticado en otra sesión.	
Acción del Actor	Respuesta del Sistema
	7a.1. Comprueba en la BD que existe otra sesión abierta para este usuario en la aplicación.
	7a.2. Muestra un mensaje indicando que el usuario ya está autenticado en la aplicación.
Referencia	R1

Poscondiciones

No aplica.

CU Cargar Configuración

Propósito

Cargar los elementos de configuración del sistema definidos para el área de trabajo del usuario autenticado.

Descripción

A partir de un usuario autenticado, se obtienen de la BD con el código del área de trabajo, los datos que este utilizará.

Diseño de Interfaz de Usuario

Ver anexo 3.

Precondiciones

No aplica.

Tabla de Evento

Flujo Básico	
Acción del Actor	Respuesta del Sistema
	1. Obtiene de la BD, usando el código del área de trabajo, la vista del mapa asignada a dicha área.
	2. Obtiene de la BD, usando el código del área, los siguientes datos de los recursos que se manipulan desde la misma: <ul style="list-style-type: none"> • Coordenadas. • Icono.
	3. Habilita las opciones: <ul style="list-style-type: none"> • “Acercar Mapa”. • “Alejar Mapa”. • “Ver Vista completa de Capa”. • “Centrar Mapa”. • “Seleccionar Región”. • “Ver Información de Región”.

	<ul style="list-style-type: none"> • “Mostrar Leyenda”. • “Identificar Elemento”. • “Abrir Capa”. • “Cerrar Capa”. • “Manipular Capa”. • “Localizar Elemento”. • “Mostrar Elemento”. • “Indicar Solicitud de Recurso”. • “Mostrar Recursos”. • “Ver Detalles de Recurso”. • “Actualizar Posición de Recursos”. • “Actualizar Estado de Recursos”. • “Encuestar Velocidad de un Recurso”. • “Mostrar Solicitud”. • “Mostrar Evento”. • “Indicar Recursos de Solicitud”. • “Mostrar Procedimientos Generales”.
	4. Muestra la vista del mapa obtenida.
Referencia	R2.

Poscondiciones

No aplica.

CU Realizar Zoom

Propósito

Permite ver un área del mapa con mayor o menor grado de detalle.

Descripción

Al seleccionarse la opción “Acercar” y un punto sobre el mapa se aumenta la vista del mapa tomando como centro las coordenadas del punto seleccionado.

Al seleccionarse la opción “Alejar” y un punto sobre el mapa se aleja la vista del mapa tomando como centro las coordenadas del punto seleccionado.

Al seleccionarse la opción “Ver Vista Completa de Capa” se muestra la vista completa de todas las capas o la capa seleccionada.

Diseño de Interfaz de Usuario

Ver anexo 4, 5 y 6.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1.El Usuario selecciona una de las siguientes opciones: <ul style="list-style-type: none"> • “Acercar”. • “Alejar”. • “Ver Vista Completa de Capa”. 	2. Comprueba el tipo de selección realizada por el usuario. <ul style="list-style-type: none"> • Si es Acercar ir a la sección “Acercar”. • Si es Alejar ir a la sección “Alejar”. • Si es Ver Vista Completa de Capa ir a la sección “Vista Completa de Capa”.
Sección “Acercar”	
3. El Usuario selecciona un punto a partir del	4. Obtiene las coordenadas del punto seleccionado.

cual desea acercar el área visible del mapa.	
	5. Aumenta el área visible del mapa a partir del punto seleccionado.
	6. Centra el área visible del mapa invocando al CU Centrar Capa usando las coordenadas del punto seleccionado.
Flujo Alternativo 3a Acercar Área	
Acción del Actor	Respuesta del Sistema
3a.1 El Usuario selecciona un área específica desplazando el puntero desde un punto inicial hasta un punto final, formando un rectángulo.	3a.2. Obtiene las coordenadas de los puntos seleccionados.
	3a.3. Enfoca la vista del mapa hasta el área especificada dentro del rectángulo.
	3a.4 Centra la vista del mapa invocando al CU Centrar Mapa usando las coordenadas del punto seleccionado.
Sección "Alejar"	
3. El Usuario selecciona un punto a partir del cual desea alejar la vista del mapa.	4. Obtiene las coordenadas del punto seleccionado.
	5. Aleja la vista del mapa a partir del punto seleccionado.
	6. Centra la vista del mapa invocando al CU Centrar Mapa usando las coordenadas del punto seleccionado.
Sección "Ver Vista Completa de Capa"	
	3. Muestra la interfaz "Vista Completa de Capa".
4. El Usuario selecciona una capa o todas las	5. Muestra la vista completa de todas las capas o la

capas		capa seleccionada.
Flujo Alterno *a Cancelar		
Acción del Actor		Respuesta del Sistema
*a.1 El Usuario selecciona la opción "Cancelar".		*a.2 Cierra la ventana "Vista Completa de Capa".
Referencia	R3, R4, R5	

Poscondiciones

No aplica.

CU Centrar Mapa

Propósito

Cambiar el área visible del mapa.

Descripción

Permite centrar el mapa a partir de un punto indicado por el usuario. Las nuevas coordenadas del centro del mapa son las del punto.

Diseño de Interfaz de Usuario

No aplica.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción centrar mapa.	
2. El Usuario selecciona un punto sobre el mapa.	3. Obtiene las coordenadas del punto

	seleccionado.
	4. Muestra la vista del mapa tomando como centro las coordenadas del punto seleccionado.
Referencia	R9

Poscondiciones

No aplica.

CU Seleccionar Región

Propósito

Seleccionar una o varias regiones sobre el mapa.

Descripción

Al seleccionarse la opción “Selección Regional” o “Selección Rectangular” sobre una región del mapa se marca la región seleccionada diferenciándola del resto del mapa.

Diseño de Interfaz de Usuario

Ver anexo 7.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona una opción de selección.	2. Comprueba el tipo de selección: <ul style="list-style-type: none"> • Si la opción es Selección Regional ir a la sección “Selección Regional”. • Si la opción es Selección Rectangular ir a la

	sección "Selección Rectangular".
Sección "Selección Regional"	
3. El Usuario selecciona uno o varios puntos sobre el mapa.	4. Obtiene las coordenadas del punto(s) seleccionado.
	5. Marca la(s) región(es) obtenida(s) diferenciándolas del resto del mapa.
	6. Comprueba que hay regiones marcadas en el mapa.
	7. Habilita la opción "Ver Información de Región."
Sección "Selección Rectangular"	
3. El Usuario traza un rectángulo sobre el mapa.	4. Comprueba que existen regiones con centro dentro del rectángulo trazado.
	5. Obtiene del mapa mostrado, las regiones cuyos centros quedan en el interior del rectángulo trazado.
	6. Marca las regiones obtenidas diferenciándolas del resto del mapa.
	7. Habilita la opción "Ver Información de Región."
Referencia	R10

Poscondiciones

No aplica.

CU Ver Información de Región

Propósito

Ver información de una región seleccionada.

Descripción

Al seleccionarse la opción “Ver Información de Región” se muestran los datos de la región seleccionada previamente por el usuario a partir de los datos recogidos de la misma en cada capa mostrada en el mapa.

Diseño de Interfaz de Usuario

Ver anexo 8.

Precondiciones

Deben existir regiones seleccionadas sobre el mapa.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Ver Información de Región”.	2. Obtiene a partir del código de las regiones seleccionadas sobre el mapa, los datos almacenados en las capas mostradas.
	3. Muestra en la interfaz “Información de la Región” los datos encontrados por cada capa mostrada.
4. El Usuario selecciona la opción “Cerrar”.	5. Cierra la interfaz “Ver Información de Región”.
Referencia	R31

Poscondiciones

No aplica.

CU Manipular Capa

Propósito

Ocultar, mostrar y ordenar las capas abiertas sobre un mapa con una configuración determinada.

Descripción

Se muestra una lista con los nombres de las capas abiertas, el usuario selecciona las capas que desea mostrar, las capas sobre las que se pueden seleccionar elementos y las capas que van a mostrar toda la información almacenada en ella. El usuario puede intercambiar también el orden de profundidad de las capas. Se muestran las capas según la configuración establecida por el usuario para realizar esta operación.

Diseño de Interfaz de Usuario

Ver anexo 9.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Manipular Capas".	2. Obtiene la configuración establecida de las capas que se muestran sobre el mapa.
	3. Muestra la interfaz "Manipular Capas".
	4. Muestra en la interfaz "Manipular Capas" la configuración establecida para cada capa abierta en el mapa.
5. El Usuario selecciona la opción "Aceptar".	6. Muestra el mapa con la configuración establecida por el usuario.
	7. Cierra la interfaz "Manipular Capas".
Flujo Alternativo *a Cerrar	
Acción del Actor	Respuesta del Sistema
*a.1. El Usuario selecciona la opción "Cerrar".	*a.2. Cierra la interfaz "Manipular Capas".

Flujo Alternativo 3a Mostrar Capa	
Acción del Actor	Respuesta del Sistema
3a.1. El Usuario marca "Mostrar" sobre una capa.	3a.2. Establece en la configuración que la capa debe mostrarse. Ir a la acción 4.
Flujo Alternativo 3b Ocultar Capa	
Acción del Actor	Respuesta del Sistema
3b.1. El Usuario desmarca "Mostrar" sobre una capa.	3b.1. Establece en la configuración que la capa debe ocultarse. Ir a la acción 4.
Flujo Alternativo 3c Seleccionar en Capa	
Acción del Actor	Respuesta del Sistema
3c.1. El Usuario marca "Seleccionar" sobre una capa.	3c.2. Establece en la configuración que se pueden seleccionar elementos en la capa seleccionada. Ir a la acción 4.
Flujo Alternativo 3d No Seleccionar en Capa	
Acción del Actor	Respuesta del Sistema
3d.1. El Usuario desmarca "Seleccionar" sobre una capa.	3d.2. Establece en la configuración que no se pueden seleccionar elementos en la capa seleccionada. Ir a la acción 4.
Flujo Alternativo 3e Mostrar Datos de la Capa	
Acción del Actor	Respuesta del Sistema
3e.1. El Usuario marca "Datos" sobre una capa.	3e.2. Establece en la configuración que se deben mostrar los datos de la capa seleccionada.

	Ir a la acción 4.
Flujo Alternativo 3f No Mostrar Datos de la Capa	
Acción del Actor	Respuesta del Sistema
3f.1. El Usuario desmarca "Datos" sobre una capa.	3f.2. Establece en la configuración que no se deben mostrar los datos de la capa seleccionada. Ir a la acción 4.
Flujo Alternativo 3g Subir Nivel de una Capa	
Acción del Actor	Respuesta del Sistema
3g.1. El Usuario selecciona "Subir" una capa.	3g.2. Coloca la capa seleccionada por encima de la anterior a esta.
	3g.3. Establece en la configuración que la capa seleccionada se debe mostrar por encima de la capa anterior. Ir a la acción 4.
Flujo Alternativo 3h Bajar Nivel de una Capa	
Acción del Actor	Respuesta del Sistema
3h.1. El Usuario selecciona "Bajar" una capa.	3h.2. Coloca la capa seleccionada por debajo de la capa siguiente.
	3h.3. Establece en la configuración que la capa seleccionada se debe mostrar por debajo de la capa siguiente. Ir a la acción 4.
Flujo Alternativo 3i Adicionar Capa	
Acción del Actor	Respuesta del Sistema

3i.1. El Usuario selecciona la opción “Adicionar”.	3i.2. Invoca al CU Abrir Capa .
	3i.3. Adiciona la capa seleccionada al mapa con la configuración establecida. Ir a la acción 6.
Flujo Alternativo 3j Eliminar Capa	
Acción del Actor	Respuesta del Sistema
3j.1. El Usuario selecciona la opción “Eliminar”.	3j.2. Invoca al CU Cerrar Capa .
	3j.3. Elimina de la configuración la capa cerrada. Ir a la acción 6.
Flujo Alternativo 3k Aplicar Configuración	
Acción del Actor	Respuesta del Sistema
3k.1. El Usuario selecciona la opción “Aplicar”.	3k.2. Ir a la acción 6.
Referencia	R11, R12, R13, R14

Poscondiciones

No aplica.

CU Identificar Elemento**Propósito**

Obtener los datos de un elemento a partir de la información que contienen las capas que se están mostrando.

Descripción

Se muestran los datos de un elemento seleccionado teniendo en cuenta la información recogida del mismo en las capas mostradas.

Diseño de Interfaz de Usuario

Ver anexo 10.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Identificar Elemento".	
2. El Usuario selecciona el elemento en el mapa.	3. Obtiene de todas las capas mostradas en el mapa los datos del elemento seleccionado.
	4. Muestra en la interfaz "Información" los datos obtenidos.
5. El Usuario selecciona la opción "Cerrar".	6. Cierra la interfaz "Información".
Referencia	R15

Poscondiciones

No aplica.

CU Localizar Elemento.**Propósito**

Ubicar un elemento en el mapa y centrar el mapa sobre ese elemento.

Descripción

Se obtienen las coordenadas del elemento seleccionado y se muestra centrando el mapa a partir de las coordenadas obtenidas.

Diseño de Interfaz de Usuario

No aplica.

Precondiciones

Debe estar seleccionado un elemento dentro de la lista.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Localizar".	2. Obtiene el código y el tipo del elemento seleccionado.
	3. Comprueba que el elemento seleccionado está mostrado en el mapa.
	4. Obtiene las coordenadas del elemento seleccionado.
	5. Invoca al CU Centrar Mapa con las coordenadas obtenidas.
Flujo Alternativo 2a Elemento No Mapificado	
	2a.1. Comprueba que el elemento seleccionado no está mostrado en el mapa.
	2a.2. Invoca al CU Insertar Elemento . Ir a la acción 5.
Referencia	R16

Poscondiciones

No aplica.

CU Mostrar Elemento

Propósito

Mostrar los elementos que cumplan con los criterios de búsquedas introducidos por el usuario.

Descripción

A partir de una capa seleccionada por el usuario y el nombre de un elemento se obtienen los elementos de dicha capa que concuerdan aproximadamente con el valor introducido por el usuario.

Diseño de Interfaz de Usuario

Ver anexo 11.

Precondiciones

Debe existir un recurso seleccionado.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Buscar Elemento en el Mapa”.	2. Muestra la interfaz “Buscar Elemento” con las capas abiertas sobre el mapa.
3. El Usuario selecciona la capa del mapa sobre la que desea realizar la búsqueda.	
4. El Usuario introduce el valor de búsqueda.	
5. El Usuario selecciona la opción “Buscar”.	6. Obtiene de la capa seleccionada los elementos que al menos uno de sus datos concuerde aproximadamente con el valor introducido.
	7. Comprueba que la búsqueda obtuvo resultado.
	8. Muestra el resultado de la búsqueda.
9. El Usuario selecciona un elemento.	10. Habilita la opción “Localizar”.
11. El Usuario selecciona la opción “Localizar”.	12. Invoca al CU Localizar Elemento .
Flujo Alternativo *a Cancelar	
Acción del Actor	Respuesta del Sistema

*a.1. El Usuario selecciona la opción “Cancelar”.	*a.2. Cierra la interfaz “Buscar Elemento”.
Flujo Alterno 7a Resultados Nulos	
Acción del Actor	Respuesta del Sistema
	7a.1. Comprueba que el resultado de la búsqueda es nulo.
	7a.1. Indica al Usuario que los resultados son nulos. Ir a la acción 2.
Referencia	R17

Poscondiciones

No aplica.

CU Insertar Elemento

Propósito

Insertar un elemento en el mapa.

Descripción

A partir del código de un elemento se obtiene de la BD las coordenadas y el estado del elemento. Muestra el elemento sobre el mapa con el icono que lo identifica.

Diseño de Interfaz de Usuario

No aplica.

Precondiciones

No aplica.

Tabla de Eventos

Flujo Básico

Acción del Actor	Respuesta del Sistema
	1. Busca en las capas del mapa si el elemento está mostrado usando el código y el tipo del elemento.
	2. Comprueba que el elemento no está mostrado sobre una capa del mapa.
	3. Obtiene de la BD, usando el código y el tipo del elemento seleccionado, los siguientes datos: <ul style="list-style-type: none"> • Coordenadas. • Icono.
	4. Inserta el icono correspondiente al elemento en la capa correspondiente y en las coordenadas obtenidas y lo muestra.
Referencia	R18

Poscondiciones

No aplica.

CU Abrir Capa**Propósito**

Abrir una capa.

Descripción

Busca y muestra una lista con las capas almacenadas en la BD. El usuario selecciona una capa de la lista y se muestra en el mapa.

Si se selecciona la opción “Abrir” del menú principal se cierran todas las capas abiertas antes de abrir la capa seleccionada.

Si se selecciona la opción “Adicionar” de la interfaz Manipular Capas se adiciona la capa a las existentes.

Diseño de Interfaz de Usuario

Ver anexo 12.

Precondiciones

No aplica.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Abrir" o la opción "Adicionar".	2. Obtiene de la BD el código y nombre de todas las capas existentes que no están mostradas en el mapa.
	3. Muestra la interfaz "Adicionar Capa:" con el nombre de las capas obtenidas.
4. El Usuario selecciona la capa que desea mostrar.	
5. El Usuario selecciona la opción "Aceptar".	6. Obtiene de la BD usando el código de la capa todos sus datos.
	7. Cierra la interfaz "Adicionar Capa".
	8. Comprueba que la opción seleccionada fue "Adicionar".
	9. Muestra la capa obtenida desde la BD.
	10. Adiciona la capa a la lista de capas abiertas.
Flujo Alternativo 5a Aplicar	
Acción del Actor	Respuesta del Sistema
5a.1 El Usuario selecciona la opción "Aplicar".	5a.2 Obtiene de la BD usando el código de la capa todos sus datos.
	5a.3 Muestra la capa obtenido desde la BD.

	Ir a la acción 3.
Flujo Alterno 8a Abrir	
	8a.1. Comprueba que la opción seleccionada fue “Abrir”.
	8a.2. Para cada capa abierta se invoca al CU Cerrar Capa usando el código de cada capa abierta. Ir a la acción 9.
Flujo Alterno *a Cancelar	
*a.1 El Usuario selecciona la opción “Cancelar”.	*a.2 Cierra la interfaz “Adicionar Capa”.
Referencia	R7

Poscondiciones

No aplica.

CU Cerrar Capa

Propósito

Cerrar una capa.

Descripción

Al seleccionar la opción “Eliminar” de la interfaz “Manipular Capas” se cierra la capa seleccionada.

Diseño de Interfaz de Usuario

No aplica.

Precondiciones

Debe existir una capa seleccionada.

Tabla de Eventos

Flujo Básico

Acción del Actor		Respuesta del Sistema
1. El Usuario selecciona una capa.		
2. El Usuario selecciona la opción "Eliminar".		3. Cierra la capa seleccionada usando su código.
		4. Elimina la capa seleccionada de la lista de capas abiertas usando su código.
Referencia	R8	

Poscondiciones

No aplica.

CU Mostrar Leyenda**Propósito**

Mostrar la leyenda del mapa mostrado.

Descripción

Se muestra una lista de iconos correspondiente a cada tipo de elemento distinto que se muestre sobre el mapa con la descripción de cada uno de ellos.

Diseño de Interfaz de Usuario

Ver anexo 13.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Mostrar Leyenda".	2. Obtiene, de las capas que se muestran sobre el mapa, los siguientes datos de cada tipo de elemento

	que sobre el mapa está iconizado: <ul style="list-style-type: none"> • Icono. • Descripción.
	3. Muestra la interfaz “Leyenda con los datos obtenidos.
4. El Usuario selecciona la opción “Cerrar”.	5. Cierra la interfaz “Leyenda”.
Referencia	R6.

Poscondiciones

No aplica.

CU Indicar Recursos de Solicitud

Propósito

Mostrar los recursos asignados a una solicitud seleccionada por el usuario.

Descripción

A partir de una solicitud seleccionada por el usuario se muestra un listado de los recursos que están asignados a dicha solicitud, destacándolos en el mapa. Diseño de Interfaz de Usuario.

Diseño de Interfaz de Usuario

Ver anexo 14.

Precondiciones

Debe existir una solicitud seleccionada.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Indicar Recursos de una Solicitud”	2. Muestra la interfaz “Indicar Recurso” con los códigos de las solicitudes que están en proceso.

3. El Usuario selecciona el código de una solicitud.	
4. El Usuario selecciona la opción "Buscar".	5. Obtiene de la BD, a partir del código de la solicitud seleccionada el código de los recursos asignados a la misma.
	6. Comprueba que la búsqueda tuvo resultados
	7. Muestra el resultado de la búsqueda.
8. El Usuario selecciona la opción "Aceptar".	9. Para cada recurso obtenido busca su icono en el mapa y lo resalta.
Flujo Alternativo *a Cancelar	
*a.1. El Usuario selecciona la opción "Cancelar".	*a.2. Cierra la interfaz "Indicar Recurso".
Referencia	R19

Poscondiciones

No aplica.

CU Mostrar Solicitud

Propósito

Mostrar los datos y localizar en el mapa las solicitudes que cumplan con los criterios de búsqueda introducidos por el usuario.

Descripción

Se muestra un listado de solicitudes que cumplen con los criterios de búsqueda especificados por el Usuario. Al seleccionarse una solicitud del listado se muestra los detalles de la misma y se habilita una opción para localizarla en el mapa.

Diseño de Interfaz de Usuario

Ver anexo 15.

Precondiciones

Debe existir una solicitud seleccionada.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Buscar Solicitud”.	2. Muestra la interfaz “Buscar Solicitud” con las Categorías de los Motivos.
3. El Usuario selecciona algún criterio de búsqueda. Los criterios son los siguientes: <ul style="list-style-type: none"> • Categoría del Motivo. • Categoría del Motivo y Motivos. 	
4. El Usuario Selecciona la opción “Buscar”.	5. Comprueba que existe algún criterio de búsqueda.
	6. Comprueba que obtuvo al menos un resultado.
	7. Obtiene de la BD y muestra, a partir del criterio de búsqueda especificado, un listado de solicitudes con los siguientes datos: <ul style="list-style-type: none"> • Código • Motivo. • Estado. • Dirección.
8. El Usuario selecciona una solicitud.	9. Habilita la opción “Localizar”
	10. Obtiene de la BD y muestra usando el código de la solicitud seleccionada, los siguientes datos:

	<ul style="list-style-type: none"> • Fecha de Registro. • Estado. • Motivo. • Dirección. • Punto de Referencia. • Descripción de la Solicitud.
11. El Usuario selecciona la opción "Localizar".	12. Invoca el CU Localizar Elemento , con el código de la misma
Flujo Alternativo *a Cerrar	
	Respuesta del Sistema
*a.1. El Usuario selecciona la opción "Cerrar".	*a.2. Cierra la interfaz "Buscar Solicitud".
Flujo Alternativo 5a Criterios de Búsqueda Nulos.	
Acción del Actor	Respuesta del Sistema
	5a.1 Comprueba que los Criterios de Búsqueda son Nulos.
	5a.4 Indica al Usuario que los criterios de búsqueda son nulos. Ir a la Acción 2.
Referencia	R20

Poscondiciones

No aplica.

CU Mostrar Evento

Propósito

Mostrar los eventos que cumplan con los criterios de búsqueda introducidos por el usuario.

Descripción

Se muestra un listado de eventos que cumplen con los criterios de búsqueda especificados. Al seleccionarse un evento del listado se muestra los detalles del mismo y se habilita una opción para localizarlo en el mapa.

Diseño de Interfaz de Usuario

Ver anexo 16 y 17.

Precondiciones

Debe existir un evento seleccionado.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Mostrar Evento”.	2. Muestra la interfaz “Mostrar Evento”, con las categorías de los eventos definidas por el centro
3. El Usuario indica: <ul style="list-style-type: none"> • Categoría del Evento. • Fecha de Ocurrencia. 	
4. El Usuario selecciona la opción “Buscar”.	5. Comprueba que exista algún criterio de búsqueda.
	6. Comprueba que obtuvo resultados en la búsqueda.
	7. Obtiene y muestra de la BD a partir del criterio de búsqueda introducido, para cada evento encontrado: <ul style="list-style-type: none"> • Código. • Estado.

	<ul style="list-style-type: none"> • Categoría.
8. El Usuario selecciona un evento.	9. Habilita la opción “Localizar”.
	10. Comprueba que el evento fue informado por un organismo.
	<p>11. Obtiene y muestra de la BD a partir del código del evento seleccionado:</p> <ul style="list-style-type: none"> • Fecha de ocurrencia. • Hora de ocurrencia • Estado. • Categoría. • Descripción. • Nombre del organismo. • Cargo que ocupa la persona que solicitó el evento. • Nombre y apellidos de la persona que solicitó el evento. • Cédula de identidad de la persona que solicitó el evento. • Teléfono para contacto. • Dirección del evento. • Punto de Referencia involucrado en la dirección del evento • Detalles de la dirección.
12. El Usuario selecciona la opción “Localizar”.	13. Invoca al CU Localizar Elemento , con el código del

	evento.
Flujo Alterno *a Cerrar	
Acción del Actor	Respuesta del Sistema
*a.1. El Usuario selecciona la opción "Cerrar".	*a.2. Cierra la interfaz "Mostrar Evento".
Flujo Alterno 10a Evento Informado por una Persona	
Acción del Actor	Respuesta del Sistema
	11a.1. Comprueba que el Evento fue informado por una persona.
	<p>11a.2. Obtiene de la BD y muestra, usando el código del evento, los siguientes datos:</p> <ul style="list-style-type: none"> • Fecha de ocurrencia. • Hora de ocurrencia. • Estado. • Categoría. • Nombre y apellidos de la persona que solicitó el evento. • Cédula de identidad de la persona que solicitó el evento. • Teléfono para contacto. • Dirección del evento. • Punto de Referencia involucrado en la dirección del evento. • Detalles de la Dirección.

	Ir a la acción 12.
Flujo Alterno 5a Criterios de Búsqueda Nulos	
	5a.1 Comprueba que el criterio de búsqueda es nulo.
	5a.2 Indica al Usuario que el criterio de búsqueda es nulo. Ir a la acción 2.
Referencia	R21, R22

Poscondiciones

No aplica.

CU Indicar Solicitud de Recurso

Propósito

Mostrar la solicitud a la que está asignado el recurso seleccionado por el usuario autenticado.

Descripción

Al seleccionarse la opción de “Indicar Solicitud de Recurso” el usuario selecciona el recurso del cual desea conocer la solicitud a la que está asignado, destacándola en el mapa.

Diseño de Interfaz de Usuario

Ver anexo 18.

Precondiciones

Debe existir un recurso habilitado.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Indicar	2. Muestra la interfaz “Indicar Solicitud de Recurso” con

Solicitud de un Recurso”.	los tipos de recursos móviles que están mostrados en el mapa.
3. El Usuario selecciona el tipo de recurso.	4. Obtiene de la BD los códigos de los recursos habilitados que cumplen con el tipo seleccionado por el usuario y que estén asignados a una solicitud.
5. El Usuario selecciona el código de un recurso.	5. Obtiene de la BD y muestra, usando el código del recurso la solicitud a la que está asignado y las coordenadas.
6. El Usuario selecciona la opción “Aceptar”.	7. Cierra la interfaz “Indicar Solicitud de Recurso”.
	8. Para la solicitud obtenida busca su icono en el mapa y lo resalta.
Flujo Alterno *a Cancelar	
*a.1 El Usuario selecciona la opción “Cancelar”	*a.2 Cierra la Interfaz “Indicar Solicitud de Recurso”.
Referencia	R23

Poscondiciones

No aplica.

CU Mostrar Recurso

Propósito

Mostrar los recursos que cumplan con el criterio de búsqueda introducidos por el usuario.

Descripción

A partir de los criterios de búsqueda introducidos por el usuario se muestra un listado de los recursos. Al seleccionarse el recurso se muestra los detalles del mismo y se habilita una opción para localizarlo en el mapa.

Diseño de Interfaz de Usuario

Ver anexo 19.

Precondiciones

No aplica

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Buscar Recursos”.	2. Muestra la interfaz “Buscar Recursos”.
3. El Usuario selecciona algún criterio de búsqueda. Los criterios de búsqueda son los siguientes: <ul style="list-style-type: none"> • Nombre del Recurso, Categoría y Tipo. • Nombre del Recurso y Categoría. • Nombre del Recurso 	
4. El Usuario selecciona la opción “Buscar”.	5. Obtiene de la BD y muestra, usando los datos introducidos por el usuario, los siguientes datos: <ul style="list-style-type: none"> • Nombre. • Categoría. • Tipo.
	6. Comprueba que la búsqueda obtuvo resultados.
7. El Usuario selecciona un recurso.	8. Obtiene de la BD y muestra, usando el código del elemento seleccionado, los siguientes datos: <ul style="list-style-type: none"> • Fotos.

	<ul style="list-style-type: none"> • Otros Datos: <ul style="list-style-type: none"> ○ Atributo. ○ Valor.
	9. Habilita la opción "Localizar".
10. El Usuario selecciona la opción "Localizar".	11. Invoca al CU Localizar Recurso .
Flujo Alterno *a Cerrar	
Acción del Actor	Respuesta del Sistema
*a.1. El usuario selecciona la opción "Cerrar".	*a.2. Cierra la interfaz "Buscar Recurso"
Flujo Alterno 6a Criterios de Búsqueda Nulos	
Acción del Actor	Respuesta del Sistema
	6a.1 Comprueba que el criterio de búsqueda es nulo.
	6a.2 Indica al Usuario que el criterio de búsqueda es nulo. Ir a la acción 2.
Referencia	R24

Poscondiciones

No aplica.

CU Ver Detalles de Recurso

Propósito

Mostrar los detalles de un recurso.

Descripción

Al seleccionar la opción "Ver Detalles" se muestran los detalles del recurso indicado por el usuario.

Diseño de Interfaz de Usuario

Ver anexo 20.

Precondiciones

Debe existir un mapa abierto.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Ver detalles de Recurso”.	2. Muestra la interfaz “Ver detalles de Recurso” con el código de los recursos que están mostrados en el mapa en ese momento.
3. El Usuario selecciona el código de un recurso.	4. Obtiene de la BD y muestra, usando el código del recurso: <ul style="list-style-type: none"> • Nombre. • Categoría. • Tipo. • Foto. • Otros Detalles: <ul style="list-style-type: none"> ○ Atributo. ○ Valor.
4. El Usuario selecciona la opción “Cerrar”.	5. Cierra la interfaz “Ver Detalles de Recurso”.
Referencia	R25

Poscondiciones

No aplica.

CU Actualizar Recurso

Propósito

Actualizar la posición y el estado de los recursos mostrados en el mapa.

Descripción

Al seleccionar la opción “Actualizar Recurso” actualiza el estado y la posición de los recursos con GPS que están siendo mostrados en el mapa.

Diseño de Interfaz de Usuario

No aplica.

Precondiciones

Debe existir al menos una capa con recursos mostrados.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Actualizar Recurso”.	2. Para cada recurso mostrado en el mapa obtiene de la BD: <ul style="list-style-type: none"> • Coordenadas. • Estado del recurso.
	3. Invoca al CU Cerrar Capa para cada capa que contenga recursos con GPS.
	4 Adiciona cada capa con los recursos actualizados.
Referencia	R26,R27

Poscondiciones

No aplica.

CU Encuestar Velocidad de Recurso**Propósito**

Mostrar la velocidad de un recurso móvil.

Descripción

Al seleccionar la opción “Encuestar Velocidad” se obtiene la velocidad de un recurso seleccionado por el usuario.

Diseño de Interfaz de Usuario

Ver anexo 21.

Precondiciones

Debe existir un recurso móvil habilitado.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Encuestar Velocidad”.	2. Muestra la interfaz “Velocidad de Recurso” con el código de todos los recursos móviles habilitados.
3. El usuario selecciona el código de un recurso.	4. Con el código del recurso seleccionado obtiene de la BD y muestra, la velocidad del recurso.
5. El usuario selecciona la opción “Cerrar”.	6. Cierra la interfaz “Velocidad de Recurso”.
Referencia	R28

Poscondiciones

No aplica.

CU Mostrar Procedimiento General

Propósito

Mostrar los Procedimientos que fueron registrados a partir de los criterios de búsquedas introducidos por el Usuario.

Descripción

A partir de los criterios de búsqueda introducidos por el Usuario se muestra un listado de Procedimientos Generales que coincidan con estos criterios. Al seleccionarse un procedimiento, el Usuario podrá ver los detalles del mismo y se habilita una opción para abrir en el mapa una vista asociada en caso de que la tenga.

Diseño de Interfaz de Usuario

Ver anexo 21.

Precondiciones

Debe existir un procedimiento general seleccionado.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción “Mostrar Detalles de Procedimientos Generales”.	2. Muestra la interfaz “Buscar Procedimientos Generales”.
3. El Usuario introduce el nombre del Procedimiento General.	
4. El Usuario selecciona la opción “Buscar”.	5. Obtiene de la BD, de todos los procedimientos generales, que su nombre o parte de este coinciden con el criterio introducido, los siguientes datos: <ul style="list-style-type: none"> • Código. • Nombre del Procedimiento General. • Si tienen vista asociada.
	6. Comprueba que la búsqueda obtuvo resultados.
	7. Muestra los resultados obtenidos.
8. El Usuario selecciona un Procedimiento	9. Invoca al CU Buscar Detalles de Procedimiento

General.	General.
	10. Muestra los siguientes datos: <ul style="list-style-type: none"> • Del Procedimiento General: <ul style="list-style-type: none"> ○ Descripción. • Recursos asignados: <ul style="list-style-type: none"> ○ Nombre del tipo de recurso. ○ Dirección.
	12. Comprueba que tiene una vista asociada.
	13. Habilita la Opción “Abrir”.
14. El usuario selecciona la opción “Abrir”.	15. Invoca al CU Ver Vista de Procedimientos Generales.
Flujo Alterno *a Cerrar.	
*a.1. El usuario selecciona la opción “Cerrar”.	*a.2. Cierra la interfaz “Buscar Procedimientos Generales”.
Flujo Alterno 6a Criterios de Búsqueda Nulos	
Acción del Actor	Respuesta del Sistema
	6a.1. Comprueba que el criterio de búsqueda es nulo.
	6a.2. Indica al Usuario que el criterio de búsqueda es nulo. Ir a la acción 2.
Referencia	R29

Poscondiciones

No aplica.

CU Abrir Vista de Procedimiento General

Propósito

Abrir la vista asociada a un Procedimiento General.

Descripción

Se obtiene de la BD los datos de la vista de un procedimiento general y se muestran. Se muestran los tipos de recursos y sus cantidades definidos en el procedimiento general seleccionado.

Diseño de Interfaz de Usuario

No aplica.

Precondiciones

Debe existir un procedimiento general seleccionado.

Tabla de Eventos

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción "Abrir".	2. Comprueba que el Procedimiento General tiene una vista asociada.
	3. Obtiene de la BD y muestra a partir del código del Procedimiento General su vista asociada.
Referencia	R30

Poscondiciones

No aplica.

2.6 Conclusiones.

Con el desarrollo de este capítulo se permitió obtener una mejor comprensión del contexto a diseñar y de las características que debe tener el sistema para cumplir con los requerimientos del cliente.

Se realizó el Modelo de Dominio, se definieron los requisitos funcionales que debe cumplir el sistema, se identificaron actores y casos de uso del sistema especificando y justificando en cuales ciclos del

desarrollo del software se van a desarrollar. Se elaboró el diagrama de casos de uso del sistema donde se representan los actores y casos de uso correspondientes, así como la relación entre ellos. Se realizó una descripción expandida para cada CU.

Capítulo 3: Descripción de la Arquitectura Propuesta.

3.1 Introducción

En este capítulo se exponen los elementos fundamentales de la arquitectura que propone MapXtreme Java Edition, se pretende realizar un análisis comparativo entre las distintas configuraciones de arquitectura que propone, arribando a conclusiones de cada una de estas. Al final del capítulo se definirá cual es la configuración más adecuada para usar en el Centro de Gestión de Emergencias de Seguridad Ciudadana (171).

La arquitectura que se propone debe ser capaz de organizar los elementos estructurales que componen el sistema y sus interfaces, junto con su comportamiento y la forma en que colaboran entre ellos.

3.2 Opciones de despliegue.

MapXtreme para Java es una herramienta para la construcción de aplicaciones para la visualización de datos geográficos. Estas aplicaciones pueden ser muy útiles para la toma de decisiones y para dirigir operaciones eficazmente. Las aplicaciones desarrolladas con MapXtreme para Java son apropiadas tanto para las Intranets corporativas como para Internet.

Para el desarrollo de estas aplicaciones MapXtreme propone tres tipos de configuración para la arquitectura o despliegues: Cliente Ligero, Cliente Pesado y Cliente Intermedio. La principal diferencia entre ellas es en cómo los datos son enviados al cliente.

Cliente Ligero

En este despliegue el cliente interactúa con páginas HTML y un navegador Web. El mapa es típicamente una imagen embebida en dichas páginas. El procesamiento de las solicitudes sobre el mapa ocurre en el servidor, este tipo de despliegue es el que comúnmente se usa en el mundo ya que no requiere código Java en el lado del cliente.

Cliente Pesado

En este tipo de despliegue, el cliente descarga un applet que provee una interfaz de usuario más sofisticada en comparación con las páginas HTML. Adicionalmente, MapXtreme puede retornar datos vectoriales en lugar de una imagen. Como el tiempo de descarga aumenta para el applet este tipo de despliegue es mejor manipularlo en sistemas de Intranet, donde el lado del cliente puede ser mejor controlado, pero además, se requiere código Java.

Cliente Intermedio

En el medio de Cliente Ligero y Cliente Pesado está el Cliente Intermedio. Al igual que un cliente pesado, el cliente intermedio descarga un applet, así que el cliente también debe tener soporte para Java. Como un cliente ligero, el cliente intermedio recibe el mapa como una imagen. El applet brinda una interfaz de usuario mucho más sofisticada en comparación con las páginas HTML y adiciona herramientas de manipulación de mapa, tales como las herramientas de selección, etc.

3.3 Vista general de MapXtreme Java Edition.

Existen cuatro componentes fundamentales que componen la Edición de MapXtreme para Java:

- 1 MapXtremeServlet.
- 2 MapJ Object.
- 3 Data Providers.
- 4 Renderers.

Veamos a continuación una breve descripción de estos componentes.

MapXtremeServlet

El MapXtremeServlet es el servidor de mapeo que provee MapXtreme para Java. Es un servlet que atiende tres tipos de solicitudes:

- Solicitudes para imágenes de mapa.
- Solicitudes para datos vectoriales de mapa
- Solicitudes para metadatos de mapa.

El MapXtremeServlet está diseñado para utilizar las capacidades del servidor web padre. Las solicitudes de imágenes son manipuladas dentro del MapXtremeServlet por un servidor de traducción multihilo (multi-threaded "Renderer server"). Similarmente, las solicitudes de datos vectoriales de mapa son manipulados por un servidor proveedor de datos también multihilo (multi-threaded "DataProvider server"). Estas características hacen que el MapXtremeServlet sea altamente escalable cuando se despliega en un servidor web. Mientras MapXtremeServlet se concentra en cumplir las solicitudes, el servidor web padre se encarga de manejar la tolerancia a fallo y la administración de la seguridad.

MapJ Object

MapJ Object es un objeto que manipula el estado del mapa. Mantiene las coordenadas del centro, el nivel de visión y las capas que lo comprenden. MapJ es el nivel más alto de la interfaz de programación de aplicaciones (API) de MapXtreme. Los MapJ Object pueden ser configurados para trabajar con diferentes Renderers y Data Providers. En la configuración más típica, MapJ es un cliente de MapXtremeServlet.

MapJ envía las solicitudes a una instancia de MapXtremeServlet y como parte del pedido, este último, suministra el servlet con el estado en curso.

Un MapJ Object puede viajar por toda la aplicación y obtener las imágenes y los datos vectoriales de mapa desde el servlet devuelto por MapXtremeServlet. Debido a que el propósito fundamental del MapJ Object es mantener el estado del mapa el consumo de memoria es muy pequeño.

Data Providers

Los Data Providers son el enlace clave entre el MapJ Object y los datos del mapa. Cada capa que conforma el mapa tiene su propio Data Provider interno. Estos son usados para acceder a los orígenes de datos y retornar datos vectoriales.

Renderers

Los Renderers son objetos que permiten la visualización de los datos sobre el mapa. Llevan a cabo el proceso de generación del mapa en forma de imagen, es decir, el paso final para crear un mapa básico es dar una imagen al cliente. A este proceso se le llama **Rendering**.

3.4 Biblioteca de etiquetas personalizadas.

MapXtreme provee una biblioteca de etiquetas personalizadas que pueden ser insertadas en las páginas JSP. Estas etiquetas se muestran como íconos permitiendo la interacción entre el usuario y el sistema, llevando a cabo la lógica del negocio de la aplicación.

Estas etiquetas han sido diseñadas para usarse en arquitecturas JSP-Servlet basada fundamentalmente en el patrón arquitectónico Modelo-Vista-Controlador (MVC). Este patrón permite separar los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Las páginas web (Vista) contienen los formularios, los cuales se envían a un servlet genérico (Controlador). El servlet controlador direcciona las peticiones a las clases apropiadas (Modelo)

Veamos como MapXtreme define estos componentes.

Vistas

Son básicamente páginas web que muestran al usuario el estado de la aplicación, redirigiendo las acciones que se realizan sobre la interfaz al controlador. Para construir las vistas, MapXtreme proporciona varias clases bases, cuyo uso depende de la funcionalidad deseada de la etiqueta. La raíz de todas estas clases es MITag, la cual implementa la interface `javax.servlet.jsp.tagext.Tag` definida en la especificación de JSP. Otras clases, de las cuales se puede heredar para obtener alguna funcionalidad específica en la aplicación, son: MapTooTag, permite mantener el foco en una funcionalidad seleccionada previamente,

MIBodyTag, permite crear etiquetas que representan los formularios de las páginas y TooTag permite crear una herramienta en la barra de herramientas para acceder a cualquier funcionalidad definida en el SIG.

Modelos

Gestionan el comportamiento y los datos de la aplicación, responde a las peticiones que realizan las vistas sobre su estado y permite su actualización desde el controlador. Las clases modelos ejecutan la lógica del negocio deseada sobre la base de las entradas de los usuarios desde las etiquetas personalizadas. Para crear una clase modelo, simplemente se implementa la interface TagBean del paquete com.mapinfo.jsptags, la cual tiene tres métodos a implementar:

- 1 **getParameterKey**, retorna una cadena única que representa la clase modelo. Esto le permite al servlet controlador determinar correctamente que clase usar para satisfacer una petición particular.
- 2 **setServerProperties**, este método permite obtener una referencia a las configuraciones de la aplicación que se definen en el fichero de configuración de la aplicación. Es llamado antes de cualquier solicitud del cliente.
- 3 **proces**, este método debe realizar la lógica del negocio necesaria. Recibe como parámetros un objeto MappingSession, un objeto HttpServletRequest y un objeto HttpServletResponse. El primer parámetro permite almacenar y mantener cualquier tipo de objeto en una sesión, el segundo representa las solicitudes del usuario y el tercero representa la respuesta dada por el sistema. El valor de retorno de este método indica al servlet controlador cómo debe proceder. Retornar verdadero significa que no se pudo satisfacer la solicitud del usuario. Un valor de retorno falso significa que la solicitud se procesó con éxito y provocará que el RequestHandler desvíe al usuario a una página JSP o HTML.

Controlador

Interpreta las acciones del usuario, accediendo a las operaciones de negocio de la aplicación y modificando a partir de sus resultados el estado del modelo y la navegación entre vistas. Para tener un servlet controlador en la aplicación no se requiere escribir uno. MapXtreme define al RequestHandler como servlet controlador de etiquetas personalizadas, el cual acepta todas las peticiones entrantes, crea el objeto MappingSession adecuado y pasa el control a la clase modelo correspondiente donde serán procesadas dichas peticiones. Para llevar a cabo estas tareas, el RequestHandler buscará los parámetros

requeridos (Ver anexo 23). En ocasiones estos parámetros son enviados como campos ocultos en formularios, pero también pueden ser añadidos a la URL.

3.5 Servlets.

Los servlets son componentes de Java que son usados para extender las funcionalidades de los servidores web. Estos no tienen interfaz gráfica de usuario.

El motor de mapeo en MapXtreme es desplegado como un servlet, el MapXtremeServlet, el cual se ajusta a la arquitectura J2EE y debe correr dentro de un servidor web con certificación para dicha arquitectura.

A continuación se explica como utilizar las diferentes opciones de despliegue aplicando el patrón arquitectónico que se propone, su diferencia radica en la localización del MapJ Object.

Cliente Ligero/Servidor Pesado

Esta es la configuración más común, un navegador web en el lado del cliente y el servlet controlador con el MapJ Object en el lado del servidor.

Cuando el cliente emite una petición a través del navegador, el servidor web envía la solicitud al servlet controlador, el cual actualiza el estado del MapJ Object, luego este objeto es usado para comunicar la solicitud de mapeo al MapXtremeServlet. Si la solicitud es para una imagen, el MapXtremeServlet devolverá una imagen del mapa al servlet controlador y este puede embeber esta imagen dentro de una página HTML y devolverla al usuario.

Esta opción tiene las siguientes características:

- 1 El MapJ Object reside dentro del servlet controlador en el lado del servidor.
- 2 El MapXtremeServlet reside en el lado del servidor.
- 3 En el cliente no se requiere código Java. El cliente puede enviar peticiones HTTP y recibir páginas HTML en la respuesta.
- 4 Produce un tráfico de red mínimo: Los applets no son requeridos. Los datos vectoriales no son enviados al cliente, solo páginas HTML con imágenes embebidas. Estas imágenes típicamente tienen un tamaño entre 15-25 Kb.

La siguiente figura ilustra esta configuración:

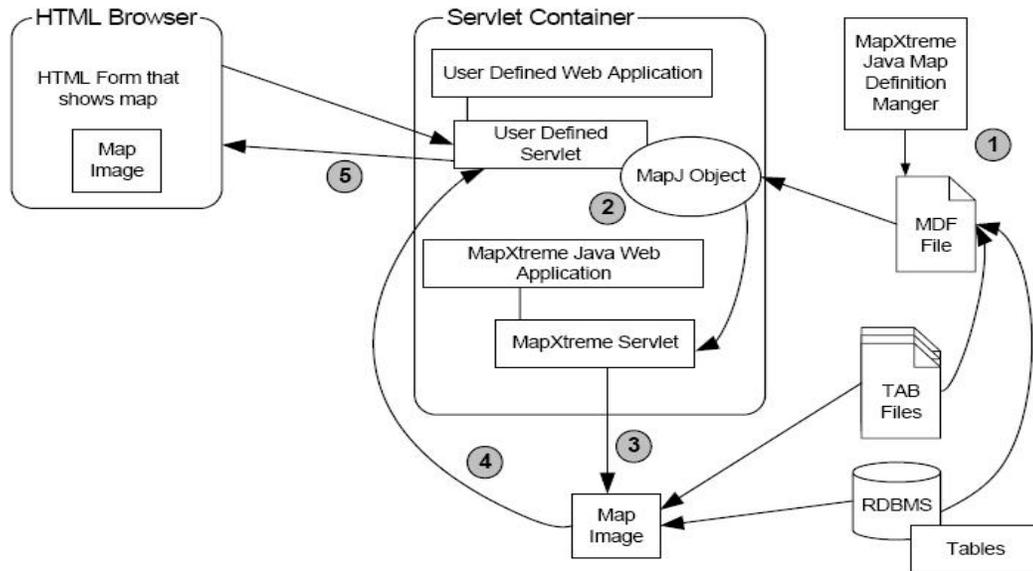


Fig. 3.1 Configuración Cliente Liger/Servidor Pesado.

Los siguientes pasos corresponden al número en el diagrama anterior.

- 1 Crear el fichero MDF en el servidor accediendo a los orígenes de datos usando Data Providers locales.
- 2 El MapJ Object creado para cada cliente conectado será inicializado usando el fichero MDF.
- 3 El servlet controlador usa el MapXtremeServlet para fabricar la imagen de mapa.
- 4 La imagen de mapa es devuelta al servlet controlador.
- 5 El servlet controlador devuelve la imagen al cliente.

La siguiente tabla destaca los puntos fundamentales de esta configuración:

Lado del cliente	Comunicación	Lado del servidor
<ul style="list-style-type: none"> • Navegador Web 	<ul style="list-style-type: none"> • Protocolo HTTP 	<p>Servlet de MapXtreme</p> <ul style="list-style-type: none"> • Generación remota del mapa usando los recursos del servidor. <p>Servlet controlador</p> <ul style="list-style-type: none"> • Acepta conexiones simultáneas.

		<ul style="list-style-type: none"> • A cada cliente conectado se le asigna un MapJ Object. • El MapJ Object se guarda en una sesión.
--	--	--

Tabla 3.1 Configuración Cliente Ligero/Servidor Pesado.

Estas características hacen que esta configuración para la arquitectura sea ideal para el despliegue en Internet, ya que tiene poco control sobre el cliente. Esta configuración puede ser usada para satisfacer a los clientes que no tienen navegadores adecuados para Java y/o tienen un ancho de banda bajo.

Cliente Pesado/Servidor Ligero

En esta configuración el MapJ Object y la lógica del negocio residen en el lado del cliente, típicamente como un Applet dentro del navegador.

La principal ventaja con este tipo de despliegue es que permite usar los JavaBeans de MapXtreme, estos JavaBeans son clases diseñadas para construir Aplicaciones Desktop. También permite que las aplicaciones puedan ser creadas mucho más rápido, además de proveer herramientas de mapa visuales, tales como: barra de herramientas y asistentes. En esta configuración, el cliente primero descarga el applet con los JavaBeans desde el servidor web.

Esta opción tiene las siguientes características:

- 1 El MapJ Object reside en el lado del cliente dentro de un applet.
- 2 El cliente requiere una programación extensiva y contiene toda la funcionalidad sobre el mapa.
- 3 El MapXtremeServlet residirá en el servidor para la generación del mapa y la adquisición de datos.
- 4 Se requiere lenguaje Java en el lado del cliente y su navegador debe tener soporte para la plataforma Java 2, Máquina Virtual 1.4.0 o superior o tener un plug-in adecuado.
- 5 Tráfico de red más pesado: El applet con los JavaBeans debe ser descargado. Los datos vectoriales deben ser enviados al cliente y el tamaño de estos datos es mucho más grande que el de las imágenes de mapa.
- 6 El archivo MDF es descargado en el lado del cliente.

La siguiente figura ilustra esta configuración:

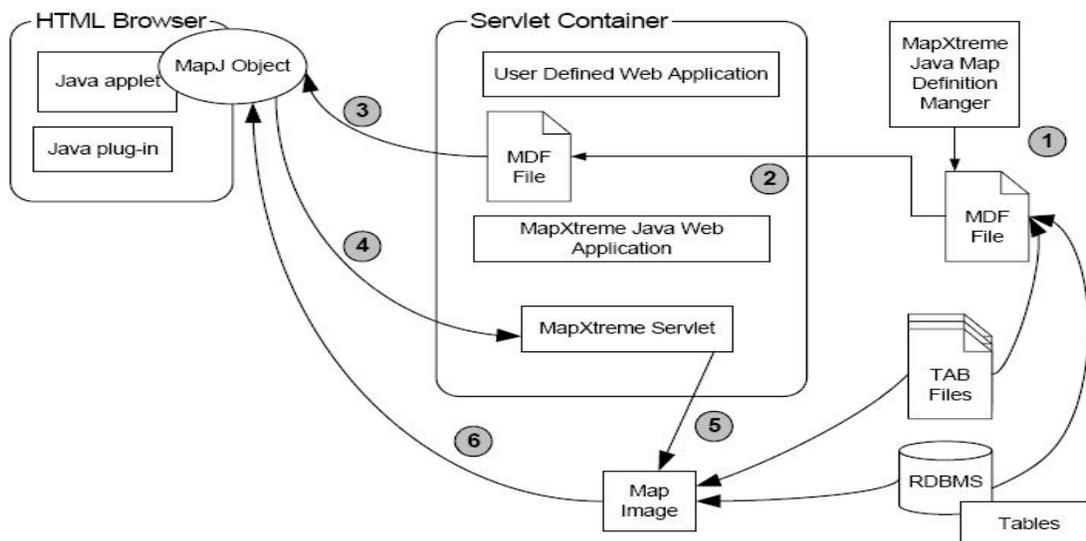


Fig 3.2 Configuración Cliente Pesado/Servidor Ligero.

Los siguientes pasos corresponden al número en el diagrama anterior.

- 1 Crear el fichero MDF en el servidor para que sea accedido localmente por los Data Providers de MapXtreme.
- 2 Poner el archivo MDF en un directorio públicamente accesible en el servidor web desde donde será consultado por el applet.
- 3 Enviar el contenido del fichero MDF desde el servidor web hasta el applet.
- 4 Enviar la configuración del MapJ Object al servidor para la creación de la imagen de mapa.
- 5 Fabricar la imagen de mapa en el servidor.
- 6 Enviar la imagen al cliente para que sea visualizada.

La siguiente tabla destaca los puntos fundamentales de esta configuración:

Lado del cliente	Comunicación	Lado del servidor
<ul style="list-style-type: none"> ▪ Navegador Web ▪ Plug-in para Applet ▪ MapJ Object 	<ul style="list-style-type: none"> ▪ Protocolo HTTP ▪ Acceso a datos remoto 	<ul style="list-style-type: none"> ▪ Generación remota del mapa usando los recursos del servidor. ▪ Adquisición de los datos a través de los Data Providers de

		<p>MapXtreme.</p> <ul style="list-style-type: none"> ▪ El archivo MDF es descargado en el cliente
--	--	--

Tabla 3.2 Arquitectura Cliente Pesado/Servidor Ligero.

Estas características hacen que esta configuración sea más conveniente para despliegues en Intranets donde el entorno es más homogéneo y controlado. Como el applet es responsable para la generación del mapa y el acceso a datos, se requiere un ancho de banda muy alto en la red. El cliente también podría requerir máquinas más poderosas.

Cliente Intermedio/Servidor Intermedio

En la configuración Cliente Intermedio/Servidor Intermedio, el applet puede ser configurado a través del servlet controlador para obtener sus datos e imágenes de mapa. Específicamente, esta configuración es usada cuando un applet se desempeña como cliente y el origen de datos es un Sistema Gestor de Bases de Datos (SGBD). En tales casos, el servlet controlador permite el acceso al SGBD.

Esta opción tiene las siguientes características:

- 1 El cliente requiere programación, aunque la modificación de los datos ocurre en el servidor.
- 2 La comunicación tiene lugar a través de objetos serializados. Esto permite que un mayor volumen de información se transmite entre el cliente y el servidor.
- 3 El MapJ Object reside en el servidor.
- 4 La generación del mapa tiene lugar en el servidor

La siguiente figura ilustra esta configuración:

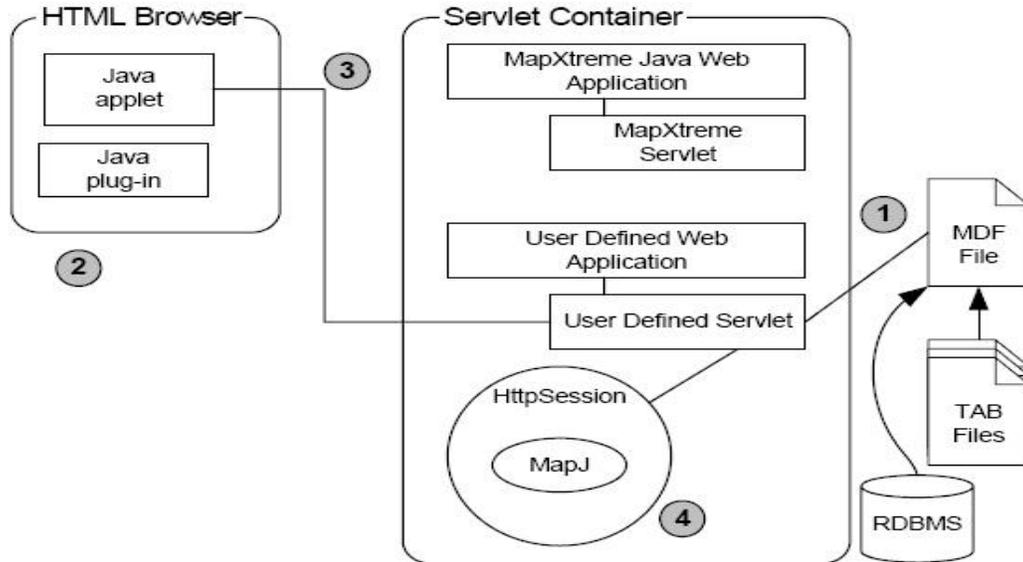


Fig 3.3 Configuración Cliente Intermedio/Servidor Intermedio.

Los siguientes pasos corresponden al número en el diagrama anterior.

- 1 El fichero MDF creado en el servidor usa Data Providers locales para cada capa.
- 2 El navegador web descarga el applet para comenzar la ejecución del mismo.
- 3 Objetos serializados de Java son usados para enviar información entre el applet y el servlet controlador.
- 4 Una sesión será creada con el servlet controlador para cada cliente conectado la cual será usada para mantener el MapJ Object del cliente.

La siguiente tabla destaca los puntos fundamentales de esta arquitectura.

Lado del cliente	Comunicación	Lado del servidor
<ul style="list-style-type: none"> ▪ Navegador Web ▪ Plug-in para Applet 	<ul style="list-style-type: none"> ▪ Protocolo HTTP ▪ Objetos serializados para el flujo de los datos 	<p>Servlet de MapXtreme</p> <ul style="list-style-type: none"> ▪ Generación remota del mapa usando los recursos del servidor. <p>Servlet controlador</p> <ul style="list-style-type: none"> ▪ Acepta conexiones simultáneas.

		<ul style="list-style-type: none"> ▪ A cada cliente conectado se le asigna un MapJ Object. ▪ El MapJ Object se guarda en una sesión.
--	--	--

Tabla 3.3 Configuración Cliente Intermedio/Servidor Intermedio.

3.6 Conclusiones.

Al concluir el capítulo nos percatamos de la necesidad de tener una arquitectura bien definida ya que esta permitirá que el sistema sea comprendido por todos los desarrolladores, permite organizar el desarrollo del mismo logrando que este evolucione positivamente. Además de que fomentará la reutilización de código.

Al finalizar la investigación decidimos que la configuración más idónea a aplicar en el Centro de Emergencia de Seguridad Ciudadana (171) es la Arquitectura Cliente Ligero/Servidor Pesado ya que esta puede ser usada por cualquier tipo de clientes.

Capítulo 4. Diseño del Sistema.

4.1 Introducción.

En este capítulo se muestran los diagramas de clases desarrollados para el módulo de Mapificación sobre Web. En estos diagramas se reflejan algunas clases que son implementados por el framework que propone MapXtreme. Los diagramas de clases de diseño muestran la especificación para las clases software de una aplicación. Incluye clases, asociaciones y atributos; Interfaces y operaciones; métodos, navegabilidad y dependencia:

4.2 Diagrama de Paquetes.

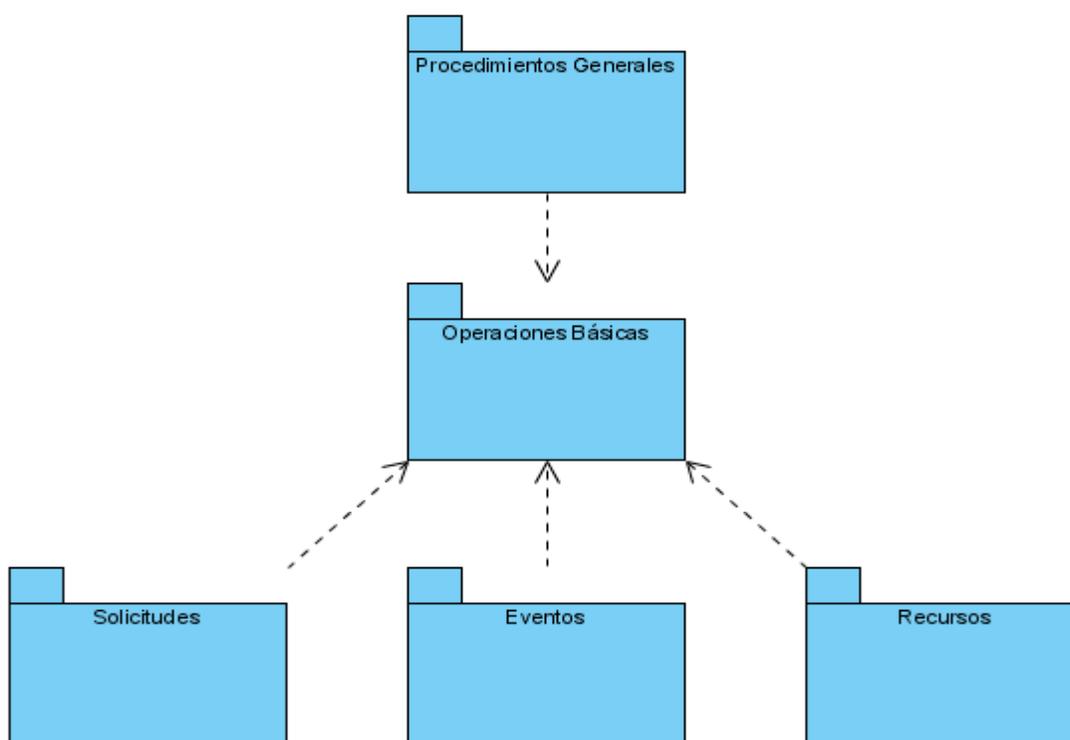


Fig 4.1 Diagrama de Paquetes.

4.3 Diagrama de Clases del Diseño.

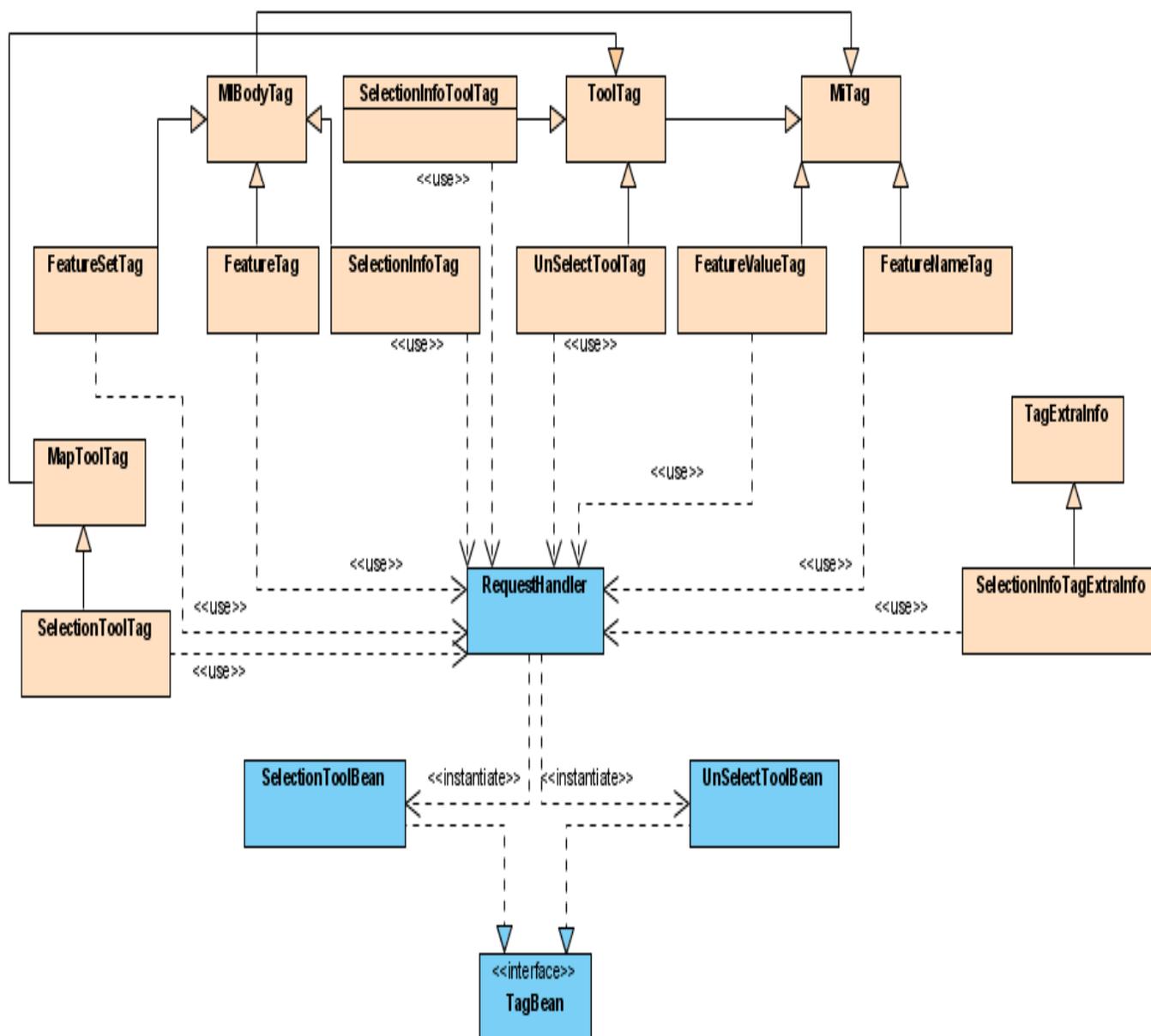


Fig 4.2 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Seleccionar Región.

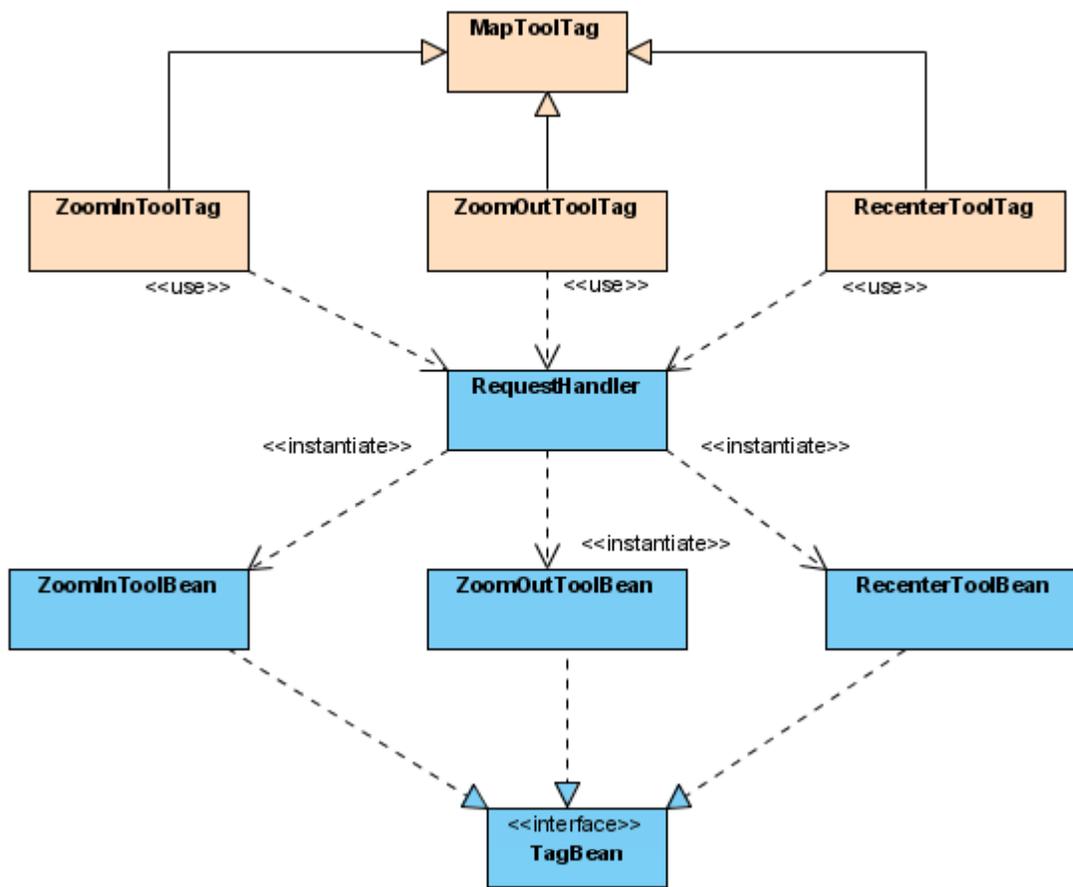


Fig 4.3 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Acercar_Alejar_Centrar Mapa.

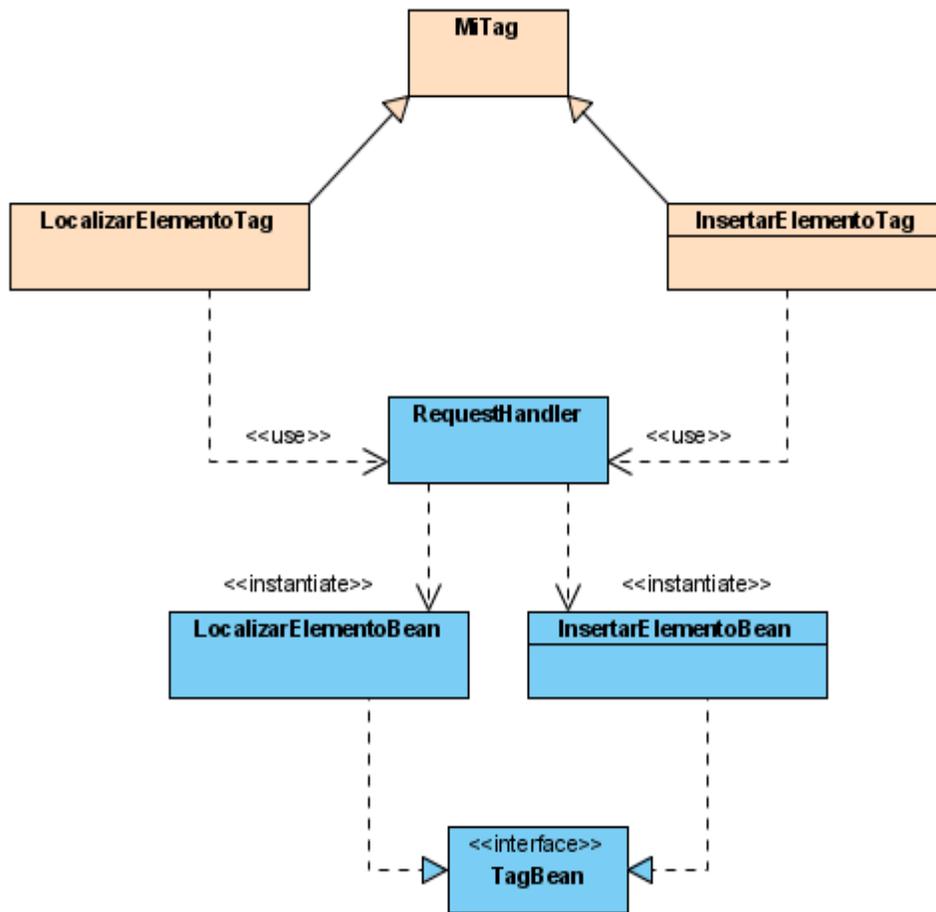


Fig 4.4 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Localizar_Insertar Elemento.

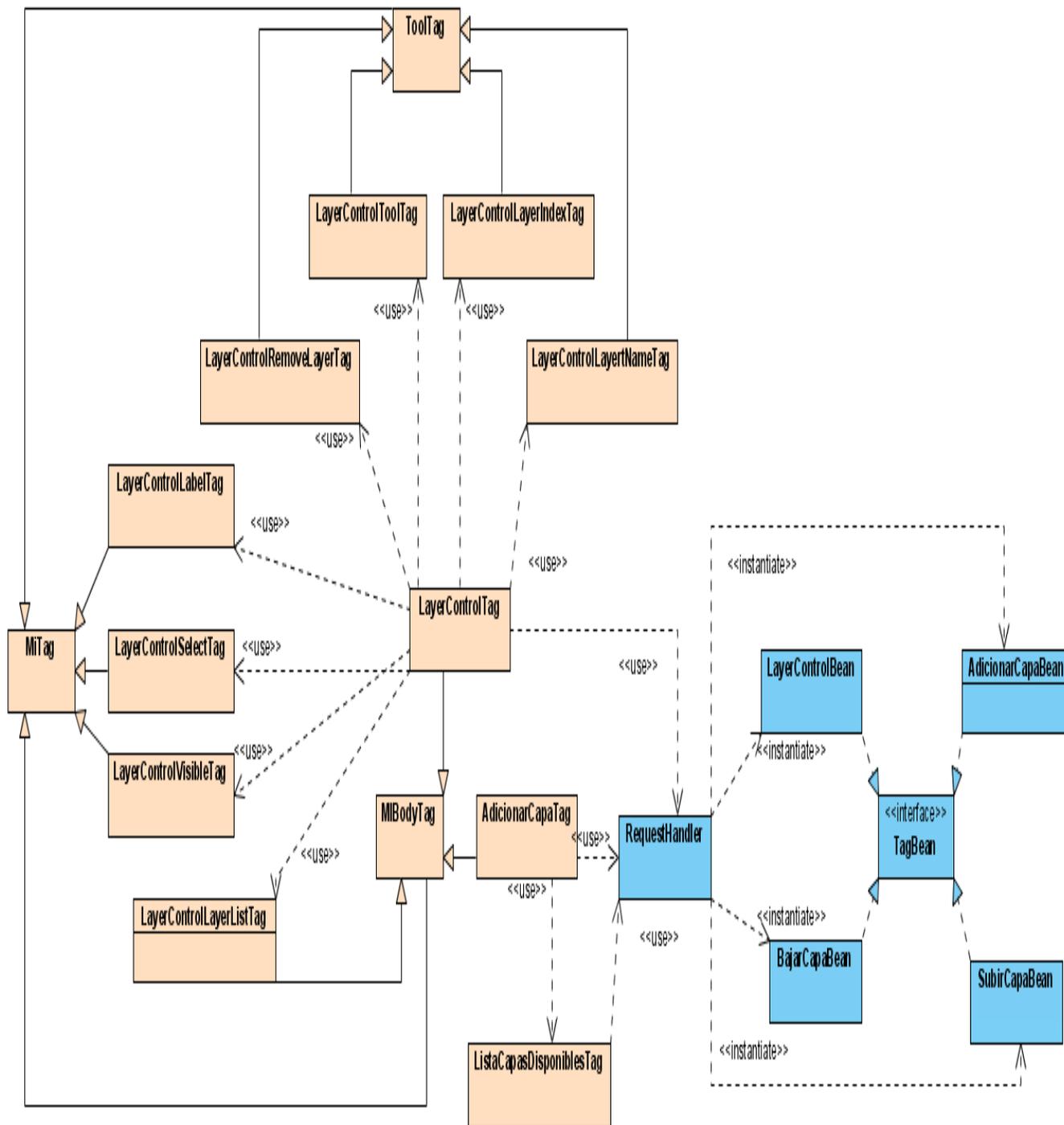


Fig 4.5 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Manipular Capa.

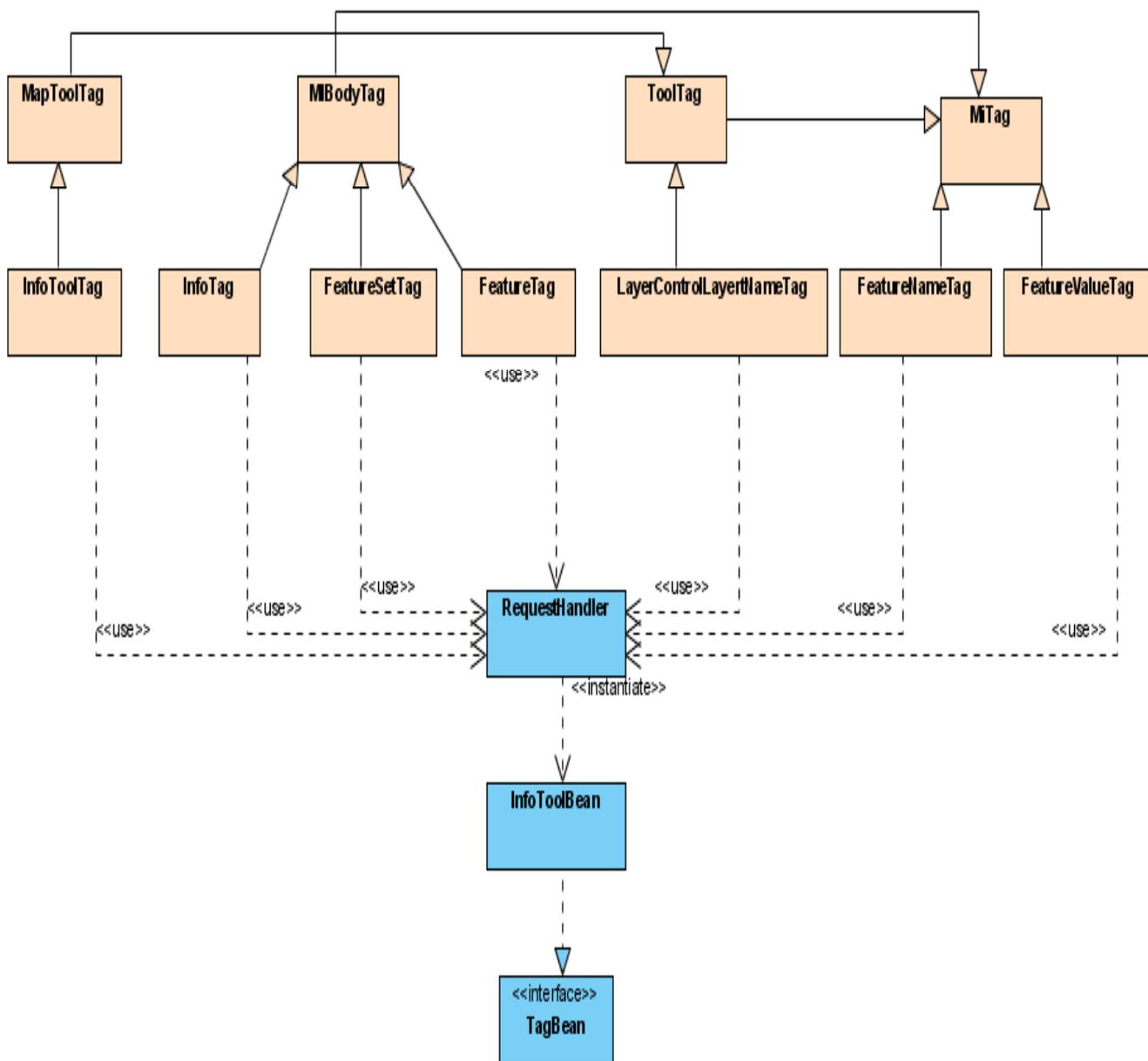


Fig 4.6 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Identificar Elemento.

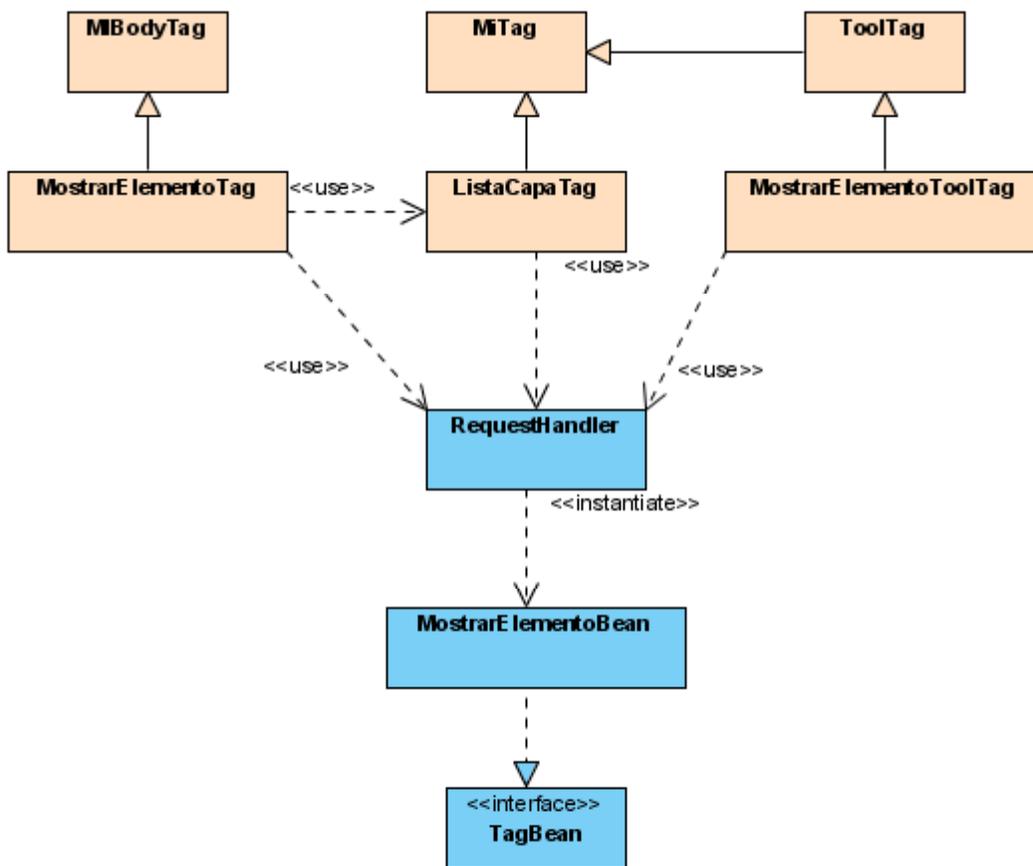


Fig 4.7 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Mostrar Elemento.

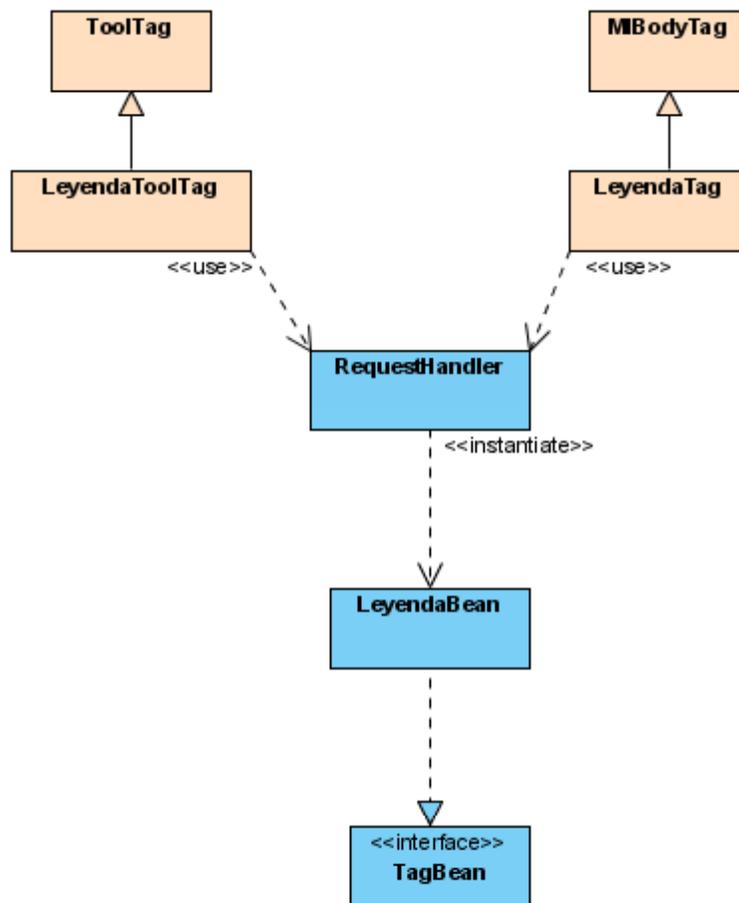


Fig 4.8 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Mostrar Leyenda.

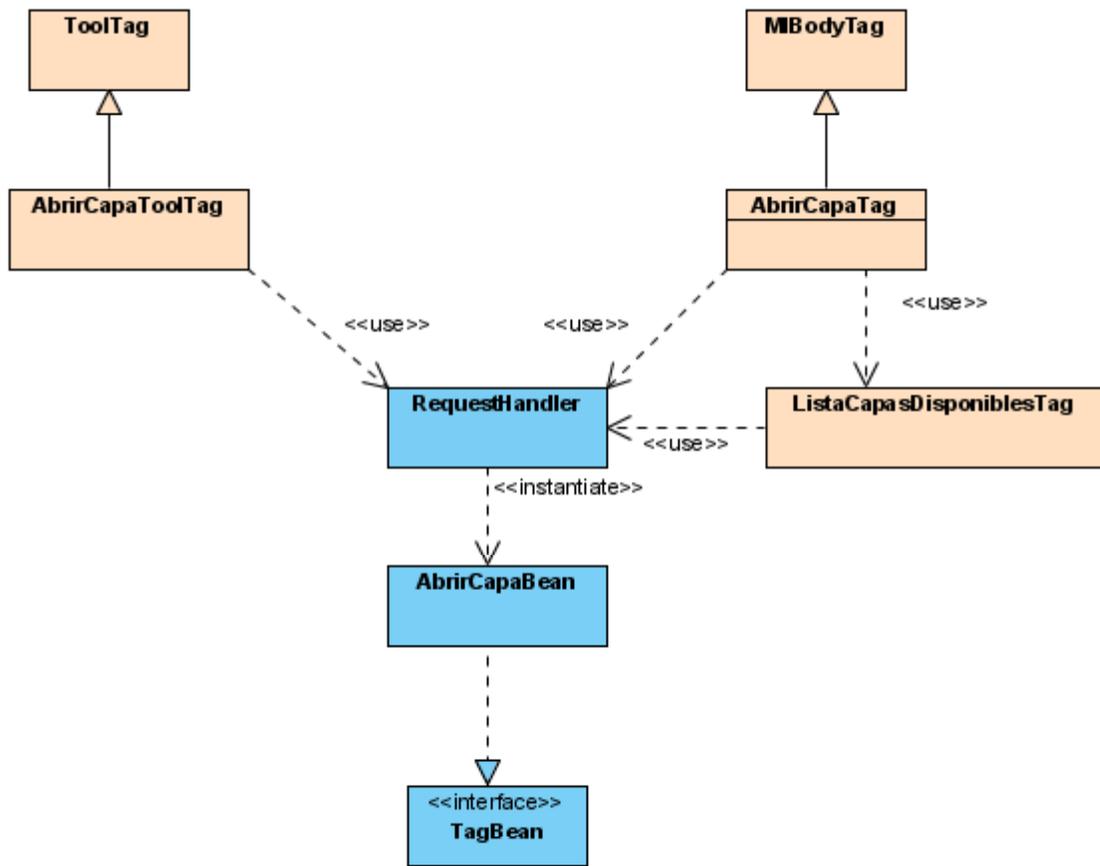


Fig 4.9 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Abrir Capa.

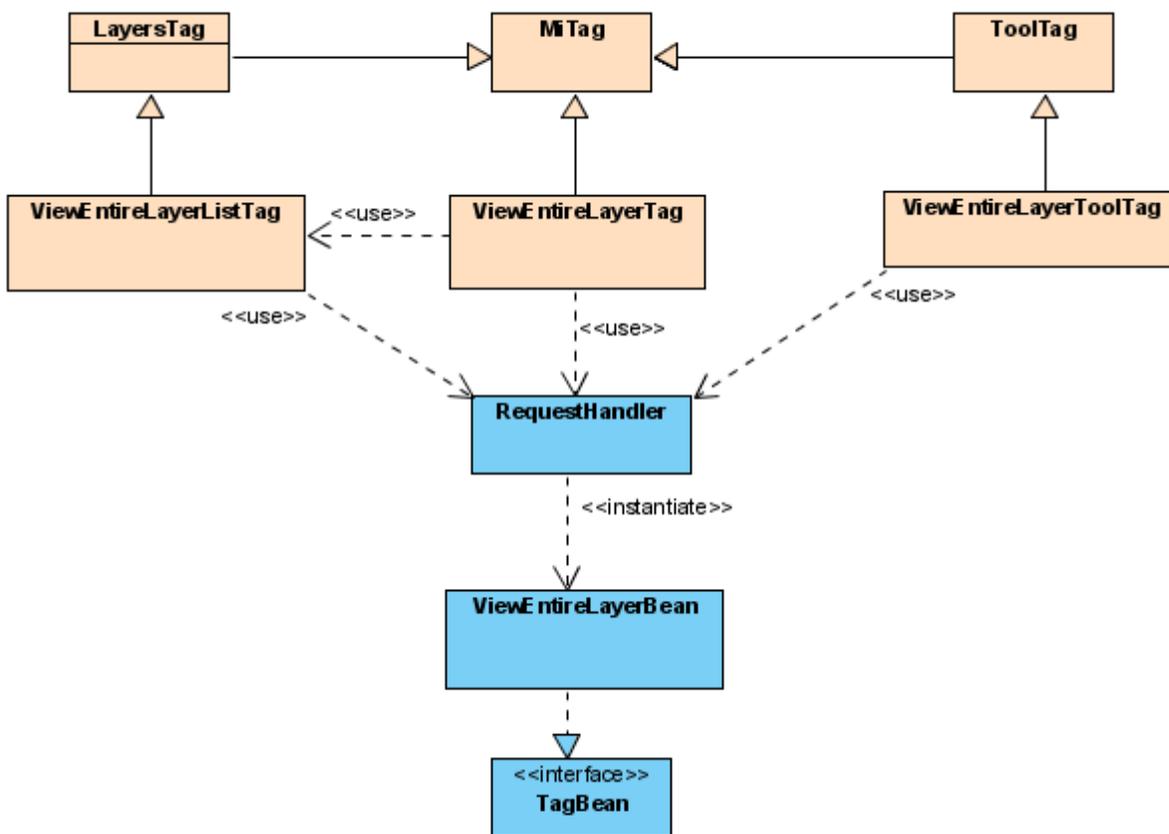


Fig 4.10 Diagrama de Clases del Diseño del Paquete Operaciones Básicas. Ver Vista Completa de Capa.

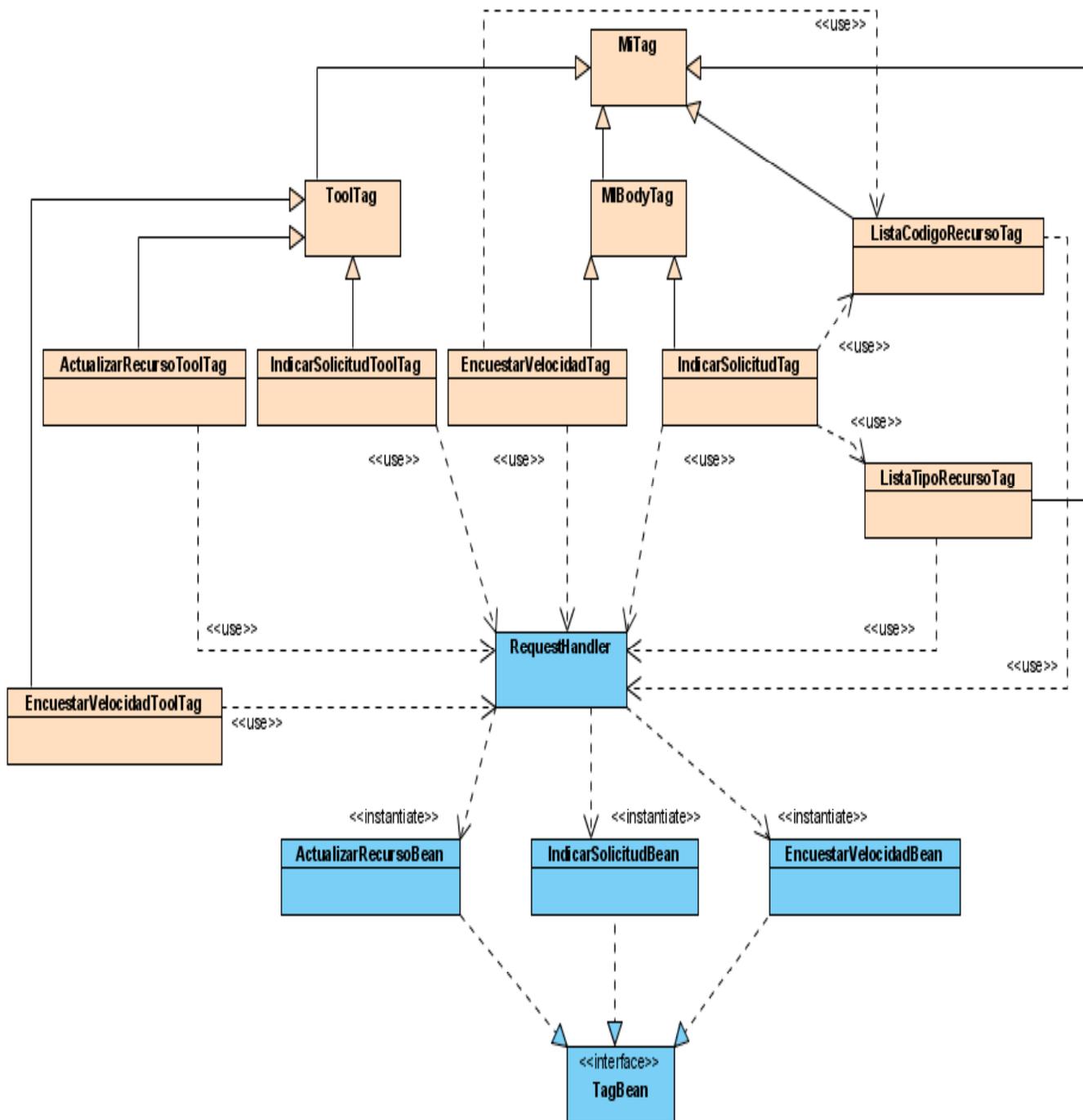


Fig 4.11 Diagrama de Clases del Diseño del Paquete Recursos. Operaciones sobre Recurso.

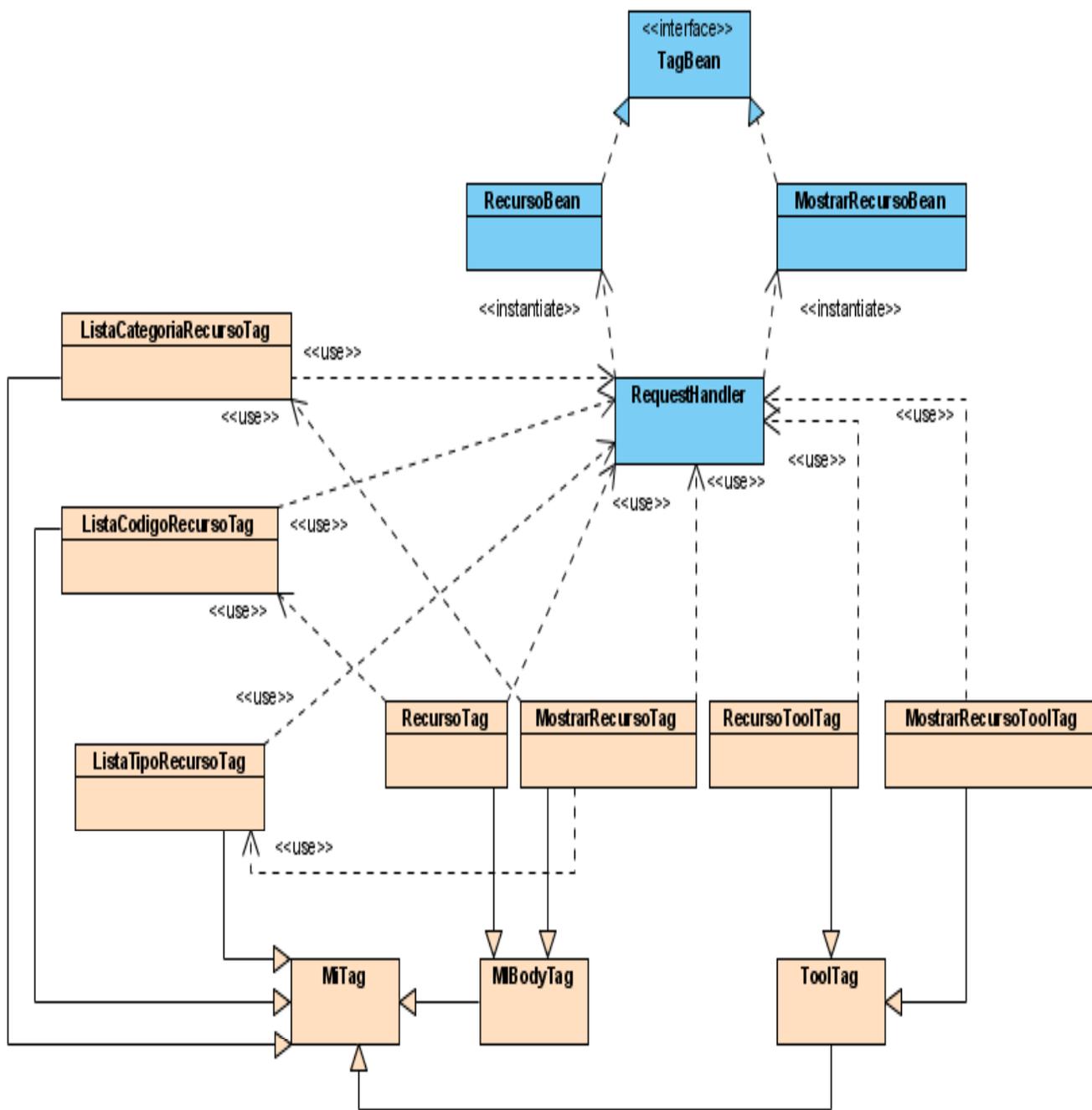


Fig 4.12 Diagrama de Clases del Diseño del Paquete Recursos. Recurso.

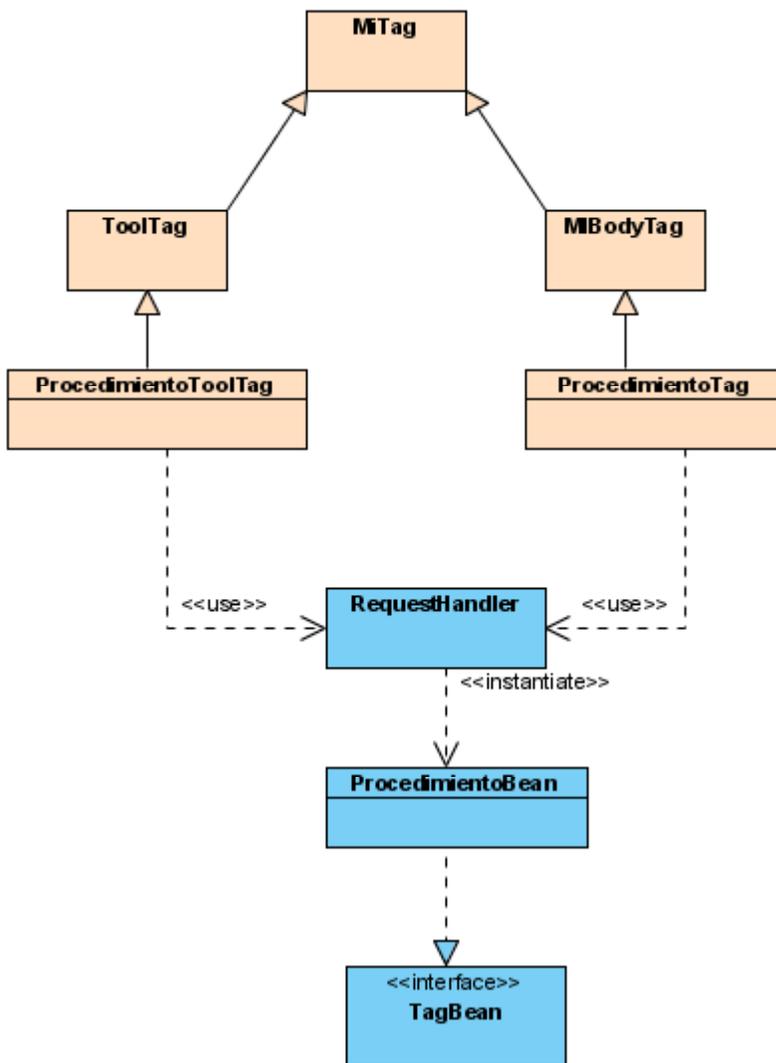


Fig 4.13 Diagrama de Clases del Diseño del Paquete Procedimientos Generales.

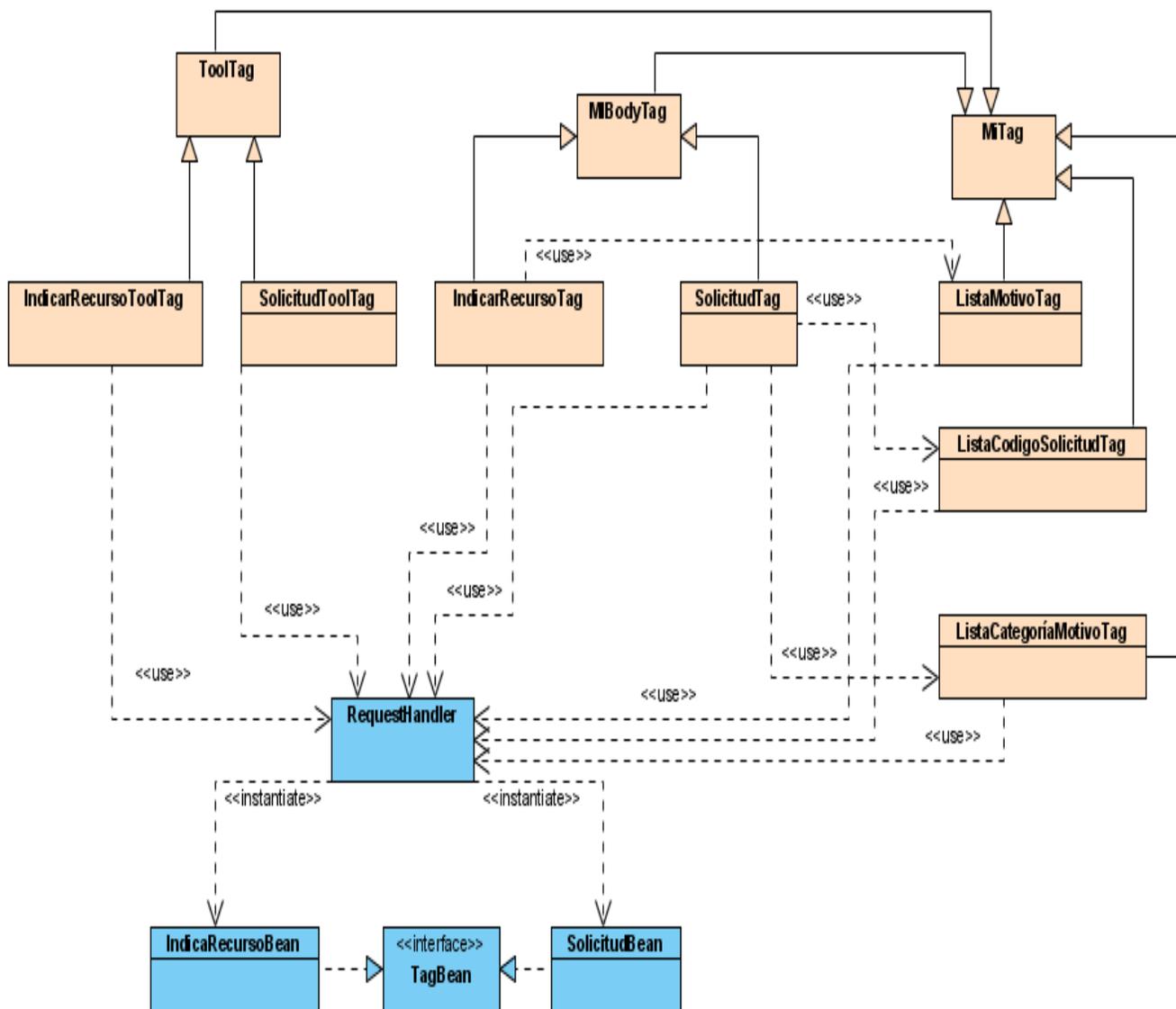


Fig 4.14 Diagrama de Clases del Diseño del Paquete Solicitudes.

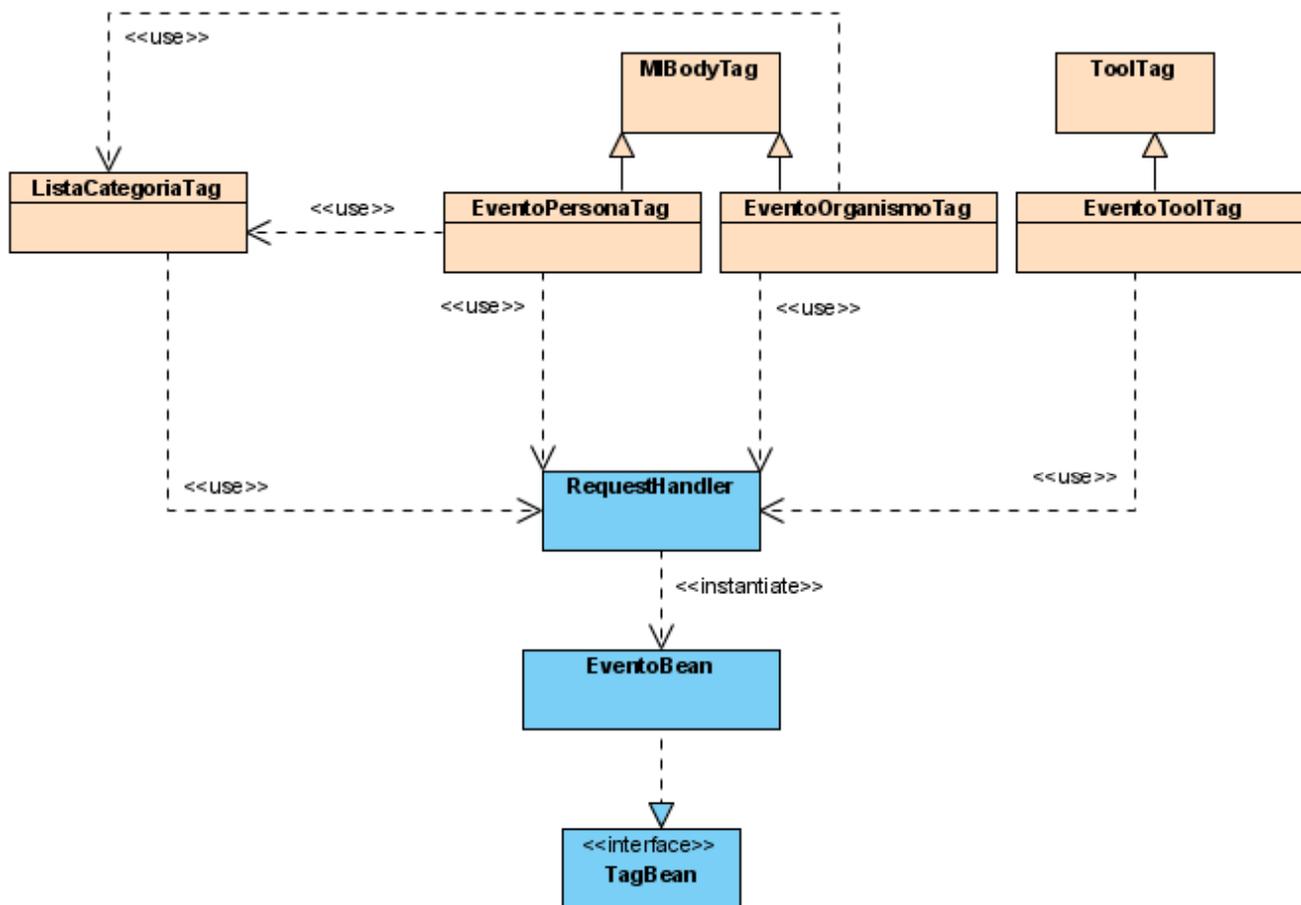


Fig 4.15 Diagrama de Clases del Diseño del Paquete Eventos.

4.4 Conclusiones.

En este capítulo se modeló detalladamente el diseño de los CU del módulo. En el anexo 24 se encuentra las clases del diseño expandidas, donde se muestran sus métodos y atributos.

Conclusiones

Al terminar la etapa de elaboración del trabajo de diploma Sistema de Gestión de Emergencias de Seguridad Ciudadana (171), “Módulo de Mapificación sobre Web” podemos afirmar que se le ha dado cumplimiento a los objetivos generales y específicos planteados al inicio del mismo.

Se realizó una investigación de las tendencias conceptuales y de software utilizados para el desarrollo exitoso de un SIG sobre web. Se seleccionó la metodología idónea para el análisis y diseño del sistema en cuestión. Y se seleccionaron las herramientas y plataforma para el desarrollo de la aplicación.

Se hizo el análisis y diseño del módulo a desarrollar tributando a la construcción de un sistema capaz de informar a los Órganos de Seguridad Ciudadana y al Centro de Gestión de Emergencias de Seguridad Ciudadana en cualquier momento que se necesite.

Recomendaciones

Realizar el análisis y el diseño de la vinculación del SIG con los reportes estadísticos que se generen en el centro mediante mapas temáticos.

Implementar la solución propuesta en este trabajo.

Referencias Bibliográficas

- [1] Venezuela, Gobierno, 2002, Gaceta Oficial de la República Bolivariana de Venezuela, Capítulo 3, Artículo 55 de la Constitución de la República Bolivariana de Venezuela.
- [2] Ortiz, Gabriel, 2005, "El funcionamiento del GPS: un repaso a los principales componentes, tipos de receptores y métodos", Fecha de acceso: 10/12/2006, Disponible en:
<http://recursos.gabrielortiz.com/index.asp?Info=039>
- [3] García Álvarez, Jose Antonio, "Así funciona el GPS", Fecha de acceso: 22/01/2007, Disponible en:
http://www.asifunciona.com/electronica/af_gps/af_gps_14.htm
- [4] Ivar Jacobson, B. C., James Runbaugh (2000) "El Proceso Unificado de Desarrollo de Software".
- [5] Corabel, S. 2004, "Manejadores de Base de Base de Datos - SQL, ORACLE, INFORMIX", Fecha de acceso: 13/03/2007, Disponible en:
<http://www.ilustrados.com/publicaciones/EpZVVlyFyAbRDtMKhI.php>
- [6] Taylor, Jennifer, 2005, "Nuevas funciones y ventajas de Dreamweaver 8", Fecha de acceso: 11/06/2007, disponible en:
http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures_02.html

Bibliografía

1. MICROSOFT, CORPORATION. Información General de Visio 2003, fecha de acceso: 9/06/2007, disponible en: <http://www.microsoft.com/latam/office/visio/prodinfo/overview.msp>
2. Equipo de Softonic, 2006, “El mejor editor HTML visual para PC”, fecha de acceso: 01/06/2007, disponible en: <http://macromedia-dreamweaver.softonic.com/>
3. Van Der Henst S., Christian, 2005, “Las novedades de Dreamweaver 8”, fecha de acceso: 01/06/2007, disponible en: <http://www.maestrosdelweb.com/actualidad/2334/>
4. Taylor, Jennifer, 2005, “Nuevas funciones y ventajas de Dreamweaver 8”, Fecha de acceso: 11/06/2007, disponible en: http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures_02.html
5. Venezuela, Gobierno, 2002, Gaceta Oficial de la República Bolivariana de Venezuela, Capítulo 3, Artículo 55 de la Constitución de la República Bolivariana de Venezuela.
6. Ortiz, Gabriel, 2005, “El funcionamiento del GPS: un repaso a los principales componentes, tipos de receptores y métodos”, Fecha de acceso: 10/12/2006, Disponible en: <http://recursos.gabrielortiz.com/index.asp?Info=039>
7. García Álvarez, Jose Antonio, “Así funciona el GPS”, Fecha de acceso: 22/01/2007, Disponible en: http://www.asifunciona.com/electronica/af_gps/af_gps_14.htm
8. Ivar Jacobson, B. C, James Runbaugh (2000) “El Proceso Unificado de Desarrollo de Software”.
9. Corabel, S. 2004, “Manejadores de Base de Base de Datos - SQL, ORACLE, INFORMIX”, Fecha de acceso: 13/03/2007, Disponible en: <http://www.ilustrados.com/publicaciones/EpZVVlyFyAbRDtMKhI.php>
10. Abián, Miguel Angel, 2003, J2EE Y .NET: LA RIVALIDAD PERMANENTE, fecha de acceso: 09/06/2007, disponible en: <http://www.javahispano.org/articles.article.action?id=55>
11. Instituto de Investigación de Recursos Biológicos “Alexander V. Humboldt”, “Los sistemas de Información Geográfica”, fecha de acceso: 07/02/2007, disponible en: <http://www.humboldt.org.co/humboldt/mostrarpagina.php?codpage=70001>
12. ArcBuk, 2005, “Introducción a los SIGs”, fecha de acceso: 01/02/2007, disponible en: <http://topografia.montes.upm.es/informacion/sig/arcbuk/arcbukmd2.html>
13. Compass/Beidou, 2006, fecha de acceso: 31/01/07, disponible en: <http://www.tecnomaps.com/index.php/2006/11/03/mas-informacion-sobre-compass-o-beidou/>
14. Garbage Collector, 2004, Sistema Gestor de Base de Datos (SGBD), fecha de acceso: 24/11/06: disponible en: http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_dato.html

15. MICROSOFT, CORPORATION, "What is SQL Server 2005?", Fecha de acceso: 24/11/2006, disponible en: <http://www.microsoft.com/sql/prodinfo/overview/what-is-sql-server.msp>
16. Mendoza Sánchez, Maria A, 2004, " Metodología de desarrollo de software ", Fecha de acceso : 20/5/2007, disponible en : http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
17. Valle Martínez, Marco A, 2002, " Seguridad Sostenible", Fecha de acceso : 06/11/2006, disponible en : <http://www.iigov.org/ss/article.drt?edi=181898&art=184288>
18. "Un concepto de Seguridad Ciudadana", fecha de acceso :04/11/2006, disponible en : http://www.senado.gob.mx/internacionales/assets/docs/relaciones_parlamentarias/america/foros/parla_latino/seguridad11.pdf
19. "Herramientas Case", fecha de acceso: 12/06/07, disponible en: <http://personales.com/cuba/cardenas/internet/case.html>
20. "Modelo Vista Controlador", fecha de acceso: 29/05/07, disponible en: <http://www.desarrollaconmsdn.com/msdn/Help/Implementacion/Implementaci%C3%B3n/Dispensador/MVC.htm>
21. MapInfo Corporation, 2006, "MapXtreme Java Edition, Developer Guide".
22. MapInfo Corporation, 2006, "MapXtreme Java Edition Documentation".