

República de Cuba



**Universidad de las Ciencias Informáticas
Facultad 2**

*“Subsistema de Administración y Control de Recursos para el
Centro de Gestión de Emergencias de Seguridad Ciudadana (171)”*

**Trabajo de Diploma
Presentado para optar por el título de
Ingeniero en Ciencias Informáticas**

**Autores: Yudeisis Beltran Hernández
Tomás Rodríguez Barrios**

Tutor: Ing. Yisel Alonso Riverón

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

Dedicatoria

A nuestros padres...

Agradecimientos

Yudeisis

A mis padres: Por apoyarme en todo, por confiar en mí y guiarme, y por ser el principal motivo de inspiración para la realización de este trabajo.

A mis hermanos: Por preocuparse siempre por mí, por ayudarme en todo.

A mis tíos Rafael y Zoraida, mis abuelos y toda mi familia: por estar siempre presentes, por su cariño y por su preocupación por mis estudios.

A Esneiker: por todo su amor, por estar siempre a mi lado, por apoyarme en todo y comprenderme.

A Yanetsy: Por preocuparse por mí y por su ayuda, que me permitió acceder a gran parte de la información necesaria para el desarrollo de este trabajo.

A Tomás: Por confiar en mí, por apoyarme y defenderme siempre.

A Yahima, Layla, Ailin, Temis, Rosa, a todo el grupo: Por darme su ejemplo, por inspirarme a hacer las cosas lo mejor posible y por toda la ayuda que me han dado.

A Yisel, Adrián, Rammel y todos los compañeros del proyecto: Porque de una forma u otra me han apoyado y se han preocupado porque todo salga bien.

Tomás

A mi mamá: que ha sabido apoyarme, a quien le debo lo que soy y ha sido protagonista de este sueño.

A mi abuela: donde quiera que esté, por todo el apoyo y la confianza que siempre depositó en mí, aún cuando no pudo verme graduado.

A Yudeisis, Elisa, Maikel, Tur, Yanet, Yani y Rammel: por compartir junto a mí en las buenas y en las malas durante estos cinco años y en el desarrollo de este trabajo.

A todos los del grupo: por la ayuda que me han prestado aún cuando nos conocimos este mismo año.

A Yisel y la dirección del proyecto: por darnos la oportunidad de pertenecer a este equipo de desarrollo en el que tanto hemos aprendido.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 5 días del mes de julio del año 2007.

Yudeisis Beltran Hernández. Tomás Rodríguez Barrios. Ing. Yisel Alonso Riverón.

Firma del Autor

Firma del Autor

Firma del Tutor

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Sistema de Gestión de Emergencias de Seguridad Ciudadana (171). Módulo de Administración y Control de Recursos.

Autor: Yudeisis Beltrán Hernández

Tomás Rodríguez Barrios

Hace ya casi un año el Sistema de Gestión de Emergencias de Seguridad Ciudadana necesitó aumentar su plantilla para poder desarrollar la solución de software cuyo alcance iba en incremento. Es así como se incorporan los estudiantes que hoy están aquí para defender el Módulo de Administración y Control de Recursos de este sistema como su trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

En este tiempo los diplomantes mostraron un alto grado de interés, dedicación e independencia en su trabajo, tomando decisiones acertadas en momentos críticos que han permitido ir obteniendo gradualmente los resultados esperados. Han sido además originales y creativos en cada una de sus soluciones, lo que ha sido extremadamente importante para enfrentarse a uno de los módulos críticos del Sistema de Gestión de Emergencias de Seguridad Ciudadana, por el proceso que resuelve y por la complejidad de la solución que permite satisfacer a plenitud los requerimientos del cliente.

Estos jóvenes; ejemplos de compromiso y protagonismo, partes de la inmensa obra que es la Universidad de las Ciencias Informáticas, pilares innegables de nuestra revolución, han contribuido con su excelente trabajo a la construcción de un sistema que incidirá en uno de los problemas de la sociedad venezolana actual, la seguridad ciudadana. Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

Ing. Yisel Alonso Riverón

Firma

20 de Junio del 2007

Fecha

Resumen

En la actualidad temas como la seguridad ciudadana y la prevención de la violencia figuran entre las principales tareas en los planes de desarrollo de países como la República Bolivariana de Venezuela. Con el propósito de garantizar la seguridad de cada ciudadano en este país se han creado los Centros de Gestión de Emergencias de Seguridad Ciudadana (171).

Actualmente existen varios de estos centros y si bien es cierto que algunos cumplen de forma eficiente su objetivo, no todos satisfacen las necesidades de la población. Para dar solución a esta problemática se propone el desarrollo de un software que estará compuesto por varios subsistemas dentro de los cuales figura uno para la administración y control de los recursos.

El principal objetivo de este trabajo es diseñar el subsistema de Administración y Control de Recursos, el cual una vez implementado permitirá administrar y controlar los recursos que los organismos ponen a disposición del centro 171, gestionará los recursos necesarios para dar solución a una situación de emergencia, permitirá conocer los recursos disponibles en cada área en los diferentes turnos de trabajo, así como los mantenimientos que necesiten los mismos. Todo ello permitirá una mayor eficacia y rapidez a la hora de dar solución a una emergencia determinada.

Este documento recoge el resultado de todo el trabajo investigativo realizado. Se propone la realización de una aplicación web utilizando la metodología de desarrollo RUP, Java como plataforma y lenguaje de programación y como herramientas de modelado, el Visual Paradigm y el Visio. Se identifican los procesos que se van a automatizar y se muestran los resultados del diseño para el subsistema de Administración y Control de Recursos propuesto.

Índice

Introducción	1
Capítulo I. Fundamentación Teórica	4
1.1 Introducción.....	4
1.2 Seguridad ciudadana	4
1.2.1 Seguridad ciudadana en Venezuela	4
1.3 Centros de gestión de emergencias	5
1.3.1 Centros de gestión de emergencias en Venezuela.....	6
1.4 Administración y control de recursos para centros de emergencias.....	7
1.5 Metodologías de desarrollo de software	7
1.5.1 Rational Unified Process (RUP)	8
1.5.2 Extreme Programming (XP)	9
1.5.3 Feature Driven Development (FDD).....	9
1.6 Selección de la metodología a utilizar	10
1.7 Herramientas Case (Computer Aided Software Engineering).....	11
1.7.1 Rational Rose	11
1.7.2 Visual Paradigm	12
1.7.3 Microsoft Office Visio.....	12
1.8 Selección de la herramienta Case a utilizar	13
1.9 Modelo IDEF0 (Integration Definition Methods)	13
1.10 Dreamweaver.....	14
1.11 Plataformas de desarrollo	15
1.11.1 Plataforma .NET	15
1.11.2 Plataforma Java	18
1.12 Selección de la plataforma y el lenguaje a utilizar	21
1.13 Conclusiones.....	21
Capítulo II. Descripción de la solución propuesta	22
2.1 Introducción.....	22
2.2 Modelo del Negocio	22
2.3 Especificación de los requisitos de software.....	24
2.4.1 Requerimientos Funcionales	24
2.4 Modelo del Sistema.....	34
2.4.1 Actores y Casos de Uso del sistema.....	34
2.4.2 Diagrama de Casos de Uso del Sistema	35
2.4.3 Descripción textual de los Casos de Uso.....	40
2.5 Conclusiones.....	95
Capítulo III. Construcción de la solución propuesta	96
3.1 Introducción.....	96
3.2 Diagramas de clases del diseño	96
3.3 Conclusiones.....	103
Conclusiones	104

Recomendaciones	105
Referencias bibliográficas	106
Bibliografía	107
Glosario de términos	109

Introducción

Los temas de la seguridad ciudadana y la prevención de la violencia han tomado gran importancia en los últimos años sumándose a las preocupaciones de los gobiernos de varios países. De esta manera, la lucha contra la inseguridad ha comenzado a figurar entre los principales temas en la agenda y planes de desarrollo de muchas naciones para asegurar la gobernabilidad democrática y la convivencia social.

Uno de los países que invierte esfuerzos en este sentido es la República Bolivariana de Venezuela, tal es así que el 20 de noviembre de 2001 se aprobó el Decreto Ley de Coordinación de Seguridad Ciudadana con el objetivo de asegurar que cada ciudadano haga valer su derecho de ser protegido físicamente de los delitos y robos que se producen en la sociedad en que se desarrolla, mediante los órganos de seguridad ciudadana, según lo establecido en el artículo 322 de la Constitución Nacional. Con el propósito de integrar a los órganos encargados de garantizar el cumplimiento de este derecho se crearon los Centros de Atención a Emergencias 171.

En la actualidad en Venezuela existen varios de estos centros destinados a brindar atención a los ciudadanos en caso de emergencias, algunos de ellos de forma bastante eficiente, pero existen otros que no brindan este servicio de manera que satisfaga todas las necesidades de la población ya que los servicios no están disponibles durante las 24 horas del día, en muchos casos no están automatizados y no siempre se dispone de los recursos necesarios para hacer frente o solucionar una situación de emergencia dada, contando con algunos estados donde no existen centros de atención.

Por otro lado en la mayoría de los casos no se tiene un control de los recursos disponibles para la atención a emergencias, ni de los terminales de comunicaciones instalados en los mismos. Todo esto provoca que la capacidad de respuesta de los órganos de seguridad ante una situación de emergencia determinada sea lenta e ineficiente, trayendo como consecuencias elevados índices de hechos delictivos o situaciones de emergencias en general, que determinan que la seguridad ciudadana se encuentre seriamente resquebrajada.

A esta situación se suma el hecho, de que los centros existentes operan de forma totalmente independiente, utilizando herramientas, formas de atención, conceptos, etc. particulares para cada uno de

ellos, es decir, no existe un estándar que defina los elementos anteriormente expuestos, o lo que es lo mismo, no se cuenta con una solución genérica que pueda aplicarse en los diferentes estados indistintamente.

Ahora bien, al enmarcarse en el estudio de los procesos de administración y control de recursos en los centros de atención a emergencias, se plantea la siguiente problemática: ¿Cómo administrar los recursos disponibles para dar una atención eficiente a las emergencias que se presenten en cada uno de los estados de la República Bolivariana de Venezuela?

Lo anteriormente planteado nos lleva a proponer un objetivo: Analizar y diseñar un subsistema genérico y replicable para la administración y control de recursos como parte del sistema informático de los Centros de Gestión de Emergencias de Seguridad Ciudadana (171) de cada uno de los estados de Venezuela.

Para llevar a cabo este proyecto investigativo se definieron un conjunto de tareas que permitirán complementar de forma satisfactoria el objetivo descrito:

- Estudio de sistemas informáticos, formas de administración y control de recursos existentes en Venezuela.
- Investigación sobre la situación a nivel nacional e internacional de las tecnologías existentes y que se pudieran utilizar para el desarrollo del proyecto.
- Seleccionar la metodología más idónea para el análisis y diseño del sistema informático que facilitará la solución del problema planteado, según las características propias del proyecto en cuestión.
- Diseñar una aplicación Web que permita gestionar los recursos que se destinan a la atención de emergencias, así como los mantenimientos asociados a los mismos.

Con el desarrollo y funcionamiento de esta aplicación se aspira agilizar y estandarizar los procesos de atención a las emergencias minimizando el tiempo de respuesta a la población, así como lograr una mayor organización en las tareas que se automatizan y con esto un riguroso control de los recursos disponibles para atender las situaciones que se presenten.

El Subsistema de Administración y Control de Recursos para el Centro de Gestión de Emergencias de Seguridad Ciudadana (171) será una aplicación genérica y replicable, que permita gestionar los recursos pertenecientes a los diferentes organismos con el fin de tener un registro de los que se encuentren disponibles, en el momento de asignarlos a la solución de una situación de emergencia determinada. Se pretende además gestionar los mantenimientos de los recursos que lo necesiten, generando un registro de los mantenimientos planificados a cada recurso para controlar su estado de disponibilidad, puesto que una vez que el recurso esté en mantenimiento no puede ser asignado a ninguna emergencia.

Mediante el uso de meta-datos el subsistema soportará la gestión y control de nuevos recursos que no se tuvieron en cuenta en un primer momento, dando mayor flexibilidad y capacidad de adaptación a dicho subsistema, permitiendo que el usuario pueda adaptarlo a sus necesidades particulares.

Este documento está estructurado en 3 capítulos.

En el Capítulo I: **Fundamentación teórica:** se definen conceptos que se abordarán durante el desarrollo del proyecto y que son de vital importancia para la comprensión del trabajo. Además se fundamentan las metodologías de desarrollo utilizadas, así como el lenguaje de programación que se utilizará.

En el Capítulo II: **Descripción de la solución propuesta:** se describen los procesos actuales de la empresa donde se aplicará el sistema a desarrollar con el objetivo de definir lo que se va a desarrollar. Se describe la solución propuesta en detalle.

En el Capítulo III: **Construcción de la solución propuesta:** se presentan modelos que describen la implementación del sistema.

Capítulo I. Fundamentación Teórica

1.1 Introducción

El desarrollo de este capítulo tiene como objetivo investigar las metodologías, lenguajes y herramientas existentes en la actualidad para diseñar el sistema. Analizar cómo se realiza el proceso de atención de emergencias teniendo en cuenta el uso de los recursos asignados para la solución de las situaciones que se presentan en la República Bolivariana de Venezuela para posteriormente plantear las características que debe tener el sistema que se desea implementar. Definir conceptos importantes para entender la investigación.

1.2 Seguridad ciudadana

Se entiende por Seguridad Ciudadana, el estado de sosiego, certidumbre y confianza que debe proporcionarse a la población, residente o de tránsito, mediante acciones dirigidas a proteger su integridad física y propiedades.[1]

Seguridad ciudadana es el esfuerzo que realiza el estado en conjunto con el pueblo con el objetivo de garantizar una situación donde se está a salvo. Se refiere a la idea de que las personas tienen derecho de sentirse protegidos de hechos que puedan producir daños tanto a su salud como a sus bienes materiales. Involucra aspectos como seguridad política, social, económica y laboral. Los gobiernos de la mayoría de los países del mundo se preocupan por prevenir, sancionar y erradicar aquellas prácticas lesivas contra las personas.

1.2.1 Seguridad ciudadana en Venezuela

En Venezuela la seguridad es un derecho que actualmente se encuentra amenazado. Cada vez más aumenta el número de homicidios, robos y secuestros que afectan principalmente a los más pobres y discriminados. Este problema es de tal magnitud que es uno de los grandes retos del gobierno venezolano

solucionarlo. Cuando se produce un hecho delictivo no se cuenta con una respuesta rápida para dar solución a estos incidentes, lo que provoca que transcurrido mucho tiempo después que se informa la situación es que llega al lugar de los hechos alguna respuesta.

Estos inconvenientes se producen actualmente debido a que no todos los centros existentes brindan sus servicios de forma que satisfaga al máximo las necesidades de la población ya que en muchos casos no están disponibles durante las 24 horas del día, otros no están automatizados y no siempre disponen de los recursos necesarios para hacer frente o solucionar una situación de emergencia.

Otro de los problemas que se presentan es que existen diversos números telefónicos para llamar en casos de emergencias y la población no cuenta con un único número fácil de recordar.

1.3 Centros de gestión de emergencias

Los centros de gestión de emergencias están disponibles para atender llamadas de emergencia de manera rápida y confiable. Estos se encargan de ayudar a las personas que se encuentran en situaciones de emergencias referidas sobre todo a asistencia médica, seguridad ciudadana e incendios, así como de brindar información necesaria para el conocimiento de las mismas.

El principal objetivo de estos centros es dar una respuesta adecuada, oportuna y eficiente mediante el llamado que realizan a los órganos destinados para resolver dichas situaciones, ya sean hospitales, bomberos o policías. Para desarrollar una labor con calidad necesitan además tener un control de todos los recursos disponibles para atender estas situaciones.

Hoy en día estos centros funcionan mediante sistemas automatizados que facilitan una atención adecuada y una respuesta rápida a las emergencias.

1.3.1 Centros de gestión de emergencias en Venezuela

En la actualidad la República Bolivariana de Venezuela cuenta con centros de gestión de emergencias en varios estados, estos se encargan de atender las llamadas que los ciudadanos realizan ante situaciones de emergencia en las que se ven involucrados. Algunos de estos centros garantizan un adecuado seguimiento de dichas situaciones, a través de los organismos competentes, prestando servicio durante las 24 horas del día. En otros centros la situación es diferente, pues solamente prestan asistencia a la población durante el horario laboral y en muchos casos ni siquiera cuentan con un sistema automatizado que agilice el proceso de atender y dar respuesta a las llamadas recibidas.

Cada uno de estos centros dispone de sus propias características y opera de forma independiente. En muchos casos no prestan los servicios suficientes para garantizar la seguridad ciudadana y no integran a los órganos de seguridad, lo que provoca que demoren su respuesta y resulten ineficientes a la hora de atender a la población.

Algunos de los centros son:

- Centro de Atención de Emergencias Aragua 171.
- Emergencias Bolívar 171.
- Sistema Integral de Emergencia de Táchira 171.
- Alcaldía Metropolitana 864-7191
- Alcaldía Libertador 545-4513 / 542-1711 / 409-8632
- Alcaldía Sucre 237-6343 / 271-0253 / 272-3360 / 0-800-76547

Como parte de esta investigación se visitaron los Centros 171 de los estados de Aragua, Bolívar y Táchira. Estados estos representativos por sus características de los tipos de emergencias que ocurren en todo el país y con diferentes maneras de administrar y controlar los recursos que están a disposición de las situaciones de emergencias a resolver.

De manera general se puede decir que la mayoría de los centros no disponen de un sistema que permita mantener un control de los recursos de forma automatizada.

1.4 Administración y control de recursos para centros de emergencias

La administración y el control de los recursos en los centros de emergencias es algo que está muy relacionado a la solución satisfactoria de las emergencias que se presentan. Este proceso implica principalmente mantener un registro de los recursos que cada organismo tiene la responsabilidad de enviar para la atención de las situaciones reportadas y de los terminales de comunicaciones instalados en los mismos. Un objetivo claro de esta actividad es mantener un control constante de los recursos de manera que a la hora de hacer frente a una situación se conozcan los que están disponibles para solucionar el problema.

Un control eficiente de los recursos permite en gran medida que se agilice el proceso de atención a las emergencias, lo cual contribuye considerablemente a elevar la calidad de las respuestas a la población.

Esto consiste principalmente en mantener actualizada la relación de móviles disponibles para la atención directa de las emergencias, así como de los lugares que puedan resultar de apoyo, ya sean hospitales, museos, centros recreativos, etc. Además de realizar reportes sobre el estado de los recursos, reparaciones, solicitudes realizadas, entre otros.

1.5 Metodologías de desarrollo de software

Actualmente los sistemas de software que se requieren son más grandes y más complejos, por lo que no es posible desarrollarlos usando los mismos métodos que se utilizaban años atrás para sistemas mucho más sencillos. Las metodologías guían los procesos de desarrollo de software donde se definen procedimientos, técnicas y herramientas que ayudan a los desarrolladores a realizar nuevos sistemas. Las metodologías de desarrollo permiten obtener software de mayor calidad que cumpla con las expectativas de los clientes. A continuación se profundizará en tres de las más conocidas.

1.5.1 Rational Unified Process (RUP)

RUP es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.[2]

Es una metodología recomendada para proyectos extensos. Divide el proceso de desarrollo de software en ciclos que terminan con una versión del producto, cada ciclo se divide en cuatro fases: inicio, elaboración, construcción y transición.

Se resume en tres características fundamentales:

Dirigido por casos de uso: Un caso de uso representa una serie de acciones que el sistema realiza para aportar un resultado al usuario. Los casos de uso representan los requerimientos funcionales. Guían el proceso de desarrollo de software ya que a partir de los casos de uso se realiza diseño, implementación y prueba del sistema.

Centrado en la arquitectura: Muestra una visión del sistema completo, describe los elementos del modelo que son más importantes para su construcción. Influye en la selección de los casos de uso significativos y estos a su vez guían la arquitectura.

Iterativo e Incremental: Para facilitar el trabajo, el proyecto se realiza mediante iteraciones que terminan con un incremento. Cada iteración comprende diferentes flujos de trabajo y de esta resulta una versión de un producto que irá creciendo en cada iteración. Esto ofrece beneficios:

- Reduce el coste de riesgo al coste de un solo incremento.
- Reduce el riesgo de no sacar el producto al mercado en la fecha prevista.
- Permite definir los requisitos del usuario a medida que se va desarrollando el producto.

1.5.2 Extreme Programming (XP)

XP es una metodología que se utiliza para proyectos pequeños, de corto plazo, con requisitos imprecisos y reducidos equipos de trabajo. Propone que entre el quipo de trabajo y el cliente haya retroalimentación continua, y que el cliente participante esté dispuesto a contestar rápidamente a las preguntas del equipo de desarrollo de forma que no se retrase la toma de decisiones. Dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.[3]

Para especificar los requisitos del software se utiliza la técnica de historias de usuarios que consiste en que el cliente describa las características que el sistema debe poseer, ya sean requisitos funcionales o no. El ciclo de vida de un proyecto aplicando esta metodología se divide en 6 fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

La programación siempre se realizará en parejas, dos personas en un mismo ordenador, cualquier programador podrá cambiar el código en cualquier parte del proyecto lo que provoca que no sea imprescindible una persona cuando se necesite cambiar el código.

1.5.3 Feature Driven Development (FDD)

Es una metodología para el desarrollo de software que se caracteriza por ser ágil, utilizada en proyectos de tiempo de duración relativamente corto. Se basa en iteraciones cortas, de aproximadamente 2 semanas que culminan con la obtención de un software funcional que el cliente y la dirección de la empresa pueden revisar.

Consiste en cinco procesos secuenciales durante los que se diseña y construye el sistema: desarrollo del modelo general, construcción de la lista de funcionalidades, planeamiento por funcionalidades, diseño por funcionalidades y construcción por funcionalidades. El trabajo se realiza en grupos donde el responsable será una persona de más experiencia que tendrá que asumir responsabilidades y tomar decisiones en caso de ser necesario.

Las clases tienen propietario, es decir, solo las puede cambiar la persona que las crea, por lo que es necesario que en un subgrupo que está implementando una funcionalidad del sistema estén presentes todos los dueños de las clases y habrá programadores que estarán presentes en varios grupos.

En el proceso de implementar la funcionalidad también se contemplan como partes del mismo la preparación y ejecución de pruebas, así como revisiones del código e integración de las partes que componen el software.[4]

1.6 Selección de la metodología a utilizar

Las metodologías analizadas tienen pocas similitudes entre sí, aunque podemos decir que XP y FDD se consideran métodos ágiles, efectivos en proyectos de corto plazo y orientados al cliente, mientras que RUP es un método pesado que se recomienda para proyectos de larga duración. En el caso de XP se genera poca documentación, lo que provoca que los requisitos del sistema final nunca se registren formalmente. En FDD el código es propietario, por lo que no puede ser modificado si no es por la persona que lo implementó.

En cuanto a la captura de requisitos, en el caso de RUP se hace mediante entrevistas con el usuario de lo que se obtienen los casos de uso, y en XP se crean las llamadas historias de usuarios, mientras que en FDD no se tiene en cuenta esta parte que es esencial para la obtención de un software con la calidad requerida.

En el caso de este proyecto se decidió utilizar la metodología RUP, por las facilidades que brinda en cuanto a la captura de requisitos, paso fundamental para el desarrollo del sistema, ya que garantiza que se obtenga un software que responda a las necesidades del cliente. Además se basa en la documentación ya que es un proyecto de larga duración, donde el equipo de desarrollo es muy inestable.

1.7 Herramientas Case (Computer Aided Software Engineering)

Las Herramientas Case son un conjunto de programas y ayudas que dan soporte a los analistas, ingenieros de software y desarrolladores durante todo el ciclo de desarrollo de un proyecto. A continuación se caracterizarán tres de las más usadas en el desarrollo de software.

1.7.1 Rational Rose

Rational Rose Enterprise Edition es una herramienta que se usa para la representación gráfica de los modelos que se obtienen durante el desarrollo de un software empleando como notación el lenguaje de modelado UML (Unified Modeling Language). Facilita la modelación de los procesos del negocio, captura de requisitos, análisis y diseño orientado a objetos, implementación del sistema mediante componentes y despliegue en las diferentes vistas: vista de casos de uso, vista lógica, vista de componentes y vista de despliegue.

Permite visualizar, entender, y refinar sus requerimientos y arquitectura antes de enfrentarse al código. Esto permite evitar esfuerzos desperdiciados en el ciclo de desarrollo. Usar una sola herramienta de modelado a través del ciclo de vida del desarrollo permite asegurar que usted está construyendo el sistema correcto.[5]

Es una herramienta compatible solamente con sistemas operativos de Microsoft que permite generar código fuente (de programas y bases de datos) a partir de los modelos para lenguajes como: Java, C++, Ada, Visual Basic, etc. e ingeniería inversa (obtención de los modelos a partir del código fuente) para diferentes lenguajes.

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

1.7.2 Visual Paradigm

Visual Paradigm es una poderosa herramienta de modelado visual que utiliza el lenguaje UML para modelar los artefactos de un proyecto de software. Soporta el ciclo de vida de un proyecto ya que facilita modelar los procesos del negocio, análisis y diseño orientado a objetos, construcción, prueba y despliegue.

Es una herramienta multiplataforma fácil de usar que permite modelar todo tipo de diagramas UML, generar documentación y código a partir de los diagramas de clases e ingeniería inversa para diferentes lenguajes de programación, como son: Java, C++, .NET, PHP entre otros. Además usando Visual Paradigm se puede obtener el diagrama entidad-relación y a partir de este el modelo de la base de datos. Es completamente compatible con Hibernate y tiene soporte para modelar EJB y generar el código de los mismos.

Presenta una excelente interoperatividad, ya que facilita importar y exportar XML así como importar archivos elaborados en Rational Rose y el modelo de datos del Erwin.

El Visual Paradigm se puede integrar con diferentes entornos de desarrollo como son: Visual Studio, Eclipse, NetBeans, Borland JBuilder, etc.

Acelera el desarrollo del software completo obteniendo productos de calidad, reduciendo costos y riesgos.

1.7.3 Microsoft Office Visio

Microsoft Office Visio es una herramienta para Windows que facilita realizar diagramas de oficinas, de procesos, de bases de datos, de flujo de programas, UML y otros. Es la herramienta adecuada para documentar, analizar y diseñar procesos empresariales complejos mediante gráficos que por lo general son mucho más fáciles de entender y que incluye mejoras en la calidad.[5]

Presenta un entorno amigable y flexible que posibilita crear diagramas de negocio para organizar y entender procesos y sistemas complejos. Mediante la representación gráfica de las ideas y relaciones en

Visio, se origina un lenguaje visual de mayor comprensión que permite llegar a soluciones satisfactorias, que llegarían a ser muy complejas usando otras herramientas.

Dispone de interfaces familiares que permiten que sea usada por una amplia gama de profesionales. Está disponible en un total de 17 idiomas, y se incluye compatibilidad mejorada para texto en idiomas asiáticos.[5]

1.8 Selección de la herramienta Case a utilizar

Basado en el estudio realizado sobre las diferentes herramientas Case, se ha seleccionado el Visual Paradigm para el modelado mediante UML de los diagramas necesarios para el desarrollo del subsistema de Administración y Control de Recursos. Se hizo esta selección teniendo en cuenta que el Visual Paradigm es una herramienta multiplataforma (está disponible para sistemas operativos de Windows, Linux, etc.), que brinda facilidades a todo el equipo de desarrollo, ya que permite generar código para diferentes lenguajes de programación e ingeniería inversa, y se integra fácilmente con IDEs como el Eclipse, el Visual Studio .NET, etc. Es una herramienta amigable, que posibilita realizar las especificaciones de los casos de uso y generar documentación sin necesidad de utilizar herramientas externas. 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.

Por otro lado se decidió utilizar Microsoft Office Visio para realizar el modelo IDEF y los prototipos de interfaz de usuario, ya que es una herramienta que permite elaborar prototipos y presenta un entorno amigable para la creación de diagramas sencillos, que posibilitan a los usuarios comprender fácilmente la información que se desea transmitir.

1.9 Modelo IDEF0 (Integration Definition Methods)

IDEF0 es un método diseñado para modelar decisiones, acciones y actividades de una organización o sistema.

Existen dos formas de modelado: AS-IS, que permite modelar los procesos actuales de la organización y TO-BE, que permite modelar cómo funcionarán los procesos con la implantación de un sistema. Estos modelos facilitan la comprensión de los procesos que se modelan. Debido a que IDEF es un lenguaje coherente y sencillo se recomienda para la modelación de procesos complejos que usando otros métodos darían lugar a diagramas confusos. En un modelo de este tipo se representan gráficamente:

Procesos: Se representan por una caja que encierra las actividades que forman parte de ese proceso.

Entradas: Representan el material o la información que es consumida o transformada por el proceso. Es posible que algunos procesos no tengan entradas.

Salidas: Material o información producida por el proceso. Cada proceso debe tener al menos una salida.

Controles: Regulan, limitan o establecen la forma en que los procesos desarrollan sus actividades para producir las salidas a partir de las entradas. Cada proceso debe tener por lo menos un control.

Mecanismos: Aquellos recursos necesarios para llevar a cabo un proceso.

Este método ayuda a los usuarios, clientes, desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el problema, para capturar correctamente los requisitos y poder construir un sistema correcto.

1.10 Dreamweaver

Dreamweaver es un editor de páginas web muy usado en el diseño y la programación web por sus funcionalidades e integración con otras herramientas. Permite al usuario utilizar la mayoría de los navegadores web, facilita la conexión a bases de datos como MySQL y SQL Server, es compatible con las tecnologías de servidor como por ejemplo, ColdFusion, ASP (Active Server Pages), ASP.NET, JSP (JavaServer Pages) y PHP.

Usando este editor el usuario puede crear páginas o sitios por medio de componentes visuales en la vista del diseño, de forma rápida y sin necesidad de escribir ningún tipo de código, es una ventaja para las personas que desean diseñar y no tiene muchos conocimientos de programación.

Dreamweaver 8 proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, características que permiten a los desarrolladores y diseñadores más o menos expertos crear rápidamente sitios web y aplicaciones basados en estándares. Desde el avanzado soporte de diseño basado en CSS a las funciones de codificación manual, provee las herramientas profesionales que requiere un entorno integrado y agilizado. Los desarrolladores pueden utilizarlo con su tecnología de servidor preferida para crear potentes aplicaciones en Internet destinadas a conectar a los usuarios a las bases de datos, las fuentes de datos dinámicos y los sistemas heredados.[6]

1.11 Plataformas de desarrollo

Una plataforma no es más que el principio de hardware o software con el cual una aplicación es compatible y permite ejecutarla. Generalmente incluye: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución.

1.11.1 Plataforma .NET

La plataforma .NET es un conjunto de tecnologías que permite a los desarrolladores utilizar diferentes lenguajes de programación desde un único entorno de desarrollo usando los mismos controles y componentes de programación. Esta plataforma surge con el objetivo de ofrecer una manera rápida, sencilla y a la vez potente de desarrollar aplicaciones. Todos estos lenguajes comparten el mismo entorno de trabajo: el Framework .NET que permite programar, compilar y ejecutar las aplicaciones.

El cual está compuesto fundamentalmente por:

El Common Language Runtime (CLR) que resulta ser el núcleo de la plataforma pues se encarga de interpretar el código que se genera en tiempo de ejecución y traducirlo al lenguaje máquina para el sistema operativo sobre el cual se está ejecutando. Proporciona funciones como la gestión de memoria, la seguridad y un sólido sistema de control de errores, a cualquier lenguaje que se integre en .NET Framework. Gracias al CLR, todos los lenguajes .NET pueden usar varios servicios de ejecución sin que

los programadores tengan que preocuparse de si su lenguaje particular admite una función de ejecución.[7]

Biblioteca de clases del Framework que proporciona un conjunto de clases orientadas a objetos e independientes del lenguaje usado, que los desarrolladores pueden utilizar para ampliar las capacidades del software. Este framework es compatible solamente con sistemas operativos de Microsoft.

1.11.1.1 C#, (o Visual C#)

C# es un nuevo lenguaje creado por Microsoft como parte de la plataforma .NET y diseñado para desarrollar aplicaciones empresariales. Presenta una serie de características que lo hacen sencillo, robusto y de alto nivel, algunas se muestran a continuación:

Orientado a Objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, que a diferencia de lenguajes como C++ no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código, soportando además todas las características propias del paradigma de programación orientada a objetos: **encapsulación, herencia y polimorfismo**.

Modernidad: se ha nutrido de elementos que se ha demostrado son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como C++ hay que simular, como un tipo básico **decimal** que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), entre otras facilidades.

Seguridad de tipos: incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

Instrucciones seguras: para evitar errores muy comunes, se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, en el caso del **switch** ha de terminar en un **break** o **goto** que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.

Sistema de tipos unificado: A diferencia de C++, todos los tipos de datos que se definan siempre se derivarán, aunque sea de manera implícita, de una clase base común llamada **System.Object**, por lo que dispondrán de todos los miembros definidos en esta clase (es decir, serán “objetos”)

Extensibilidad de tipos básicos: permite definir, a través de **estructuras**, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro **ref**.

Orientación a componentes: su propia sintaxis incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, su sintaxis permite definir cómodamente **propiedades** (similares a campos de acceso controlado), **eventos** (asociación controlada de funciones de respuesta a notificaciones) o **atributos** (información sobre un tipo o sus miembros).

Eficiencia: En principio, todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador **unsafe**) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.

Compatibilidad: Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados **Platform Invocation Services (PInvoke)**, la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que estas muchas veces esperan recibir o devuelven punteros.

Después de los elementos expuestos se puede concluir diciendo:

C# combina las mejores ideas de lenguajes como C, C++ y Java con las mejoras de productividad de .NET Framework de Microsoft y brinda una experiencia de codificación muy productiva tanto para los nuevos programadores como para los veteranos.[7]

1.11.2 Plataforma Java

La plataforma Java es una plataforma que permite crear y ejecutar aplicaciones de forma interactiva, dinámica y segura en un entorno de red. Dispone de dos componentes fundamentales:

La Máquina Virtual de Java (MVJ) que es la encargada de ejecutar los programas escritos en el lenguaje de programación Java y es quien permite que las aplicaciones desarrolladas en este lenguaje puedan ser ejecutadas en diversos sistemas con arquitecturas diferentes. El otro componente es la Interfaz de Programación de Aplicaciones de Java (API Java) que no es más que un conjunto de clases ya desarrolladas que brindan gran cantidad de posibilidades al programador como es el manejo de archivos, interfaces gráficas, acceso a Internet, acceso a bases de datos, etc.

Esta plataforma ha sido creada con la participación de diversos desarrolladores de diferentes índoles, y es una plataforma conjunta basada en el lenguaje de programación Java.

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 al ser una plataforma basada en el lenguaje Java, es posible desarrollar arquitecturas utilizando cualquier sistema operativo donde se pueda ejecutar una máquina virtual Java.

Además dispone de una amplia comunidad de desarrolladores, que enriquecen su proceso evolutivo como plataforma, contando con varios frameworks que agilizan el trabajo de los mismos.

Actualmente existen varias ediciones de la plataforma Java, cada una de ellas destinada a cubrir un conjunto diferente de necesidades de programación: J2ME (Java2 Micro Edition) usada en aplicaciones para móviles, J2EE (Java2 Enterprise Edition) usada en aplicaciones de empresas y J2SE (Java2 Standard Edition) para aplicaciones de escritorio.

1.11.2.1 Lenguaje Java

Java es uno de los lenguajes de programación más elaborados y más utilizados para la creación de software de empresa. La evolución de Java, que ha pasado de ser un medio de desarrollo de applets para ser ejecutados en navegadores a un modelo de programación capaz de manejar las aplicaciones de una empresa de hoy en día, ha sido extraordinaria.[8]

El lenguaje de programación Java surge a partir de las características de diferentes lenguajes, principalmente de C++. Es un lenguaje orientado a objetos que presenta particularidades que lo convierten en un lenguaje sencillo, seguro, estándar y de alto nivel, algunas de ellas se muestran a continuación:

Orientado a Objetos: Trabaja con sus datos como objetos y con interfaces a esos objetos. Implementa conceptos como encapsulación, herencia, abstracción, reutilización, polimorfismo, entre otros.

Distribuido: Contiene en su propio núcleo los elementos necesarios para trabajar en un entorno conectado en red. Cuenta con librerías y herramientas que permiten que los programas corran en varias máquinas interactuando.

Interpretado: Se traduce el código fuente a un código intermedio denominado ByteCodes, que es interpretado por la Máquina Virtual de Java, lo cual permite que se pueda ejecutar en cualquier sistema operativo.

Robusto: Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución pues verifica ByteCodes, maneja excepciones, evita sobrescribir o corromper memoria mediante punteros.

Seguro: el hecho de que no se permita el acceso ilegal a memoria puesto que no se utilizan punteros, brinda ya cierto nivel de seguridad; además Java utiliza otros mecanismos, dentro de los básicos se encuentran: **integración de un sistema de control de permisos para los programas** puesto que define un mecanismo (denominado *mecanismo del sandbox o caja de arena*) que se encarga de supervisar qué se le permite hacer a un programa y controlar cómo accede a los recursos. También permite el cifrado **y uso de certificados**. Se definen mecanismos para que los programadores puedan firmar el código, de manera que los usuarios puedan verificar quien es el propietario del código y que este no ha sido modificado después de ser firmado.

Arquitectura neutral: La Plataforma Java compila el código fuente a un código independiente de la arquitectura de la máquina (ByteCodes) donde se ejecutará y este es interpretado por la MVJ. Lo cual permite además que sea un lenguaje portable.

Altas prestaciones: Incorpora un compilador JIT(Just in time) que en tiempo de ejecución realiza optimizaciones según va leyendo los bytescodes y los va transformando en código de máquina, sin perder tiempo optimizando código que no se ejecutará. Además soporta el trabajo con varios microprocesadores, brindando mayor escalabilidad.

Multithread. (Multihilos): Permite la ejecución de varias tareas a la vez, proporcionando para ello un conjunto de clases que facilitan el trabajo de los desarrolladores.

1.12 Selección de la plataforma y el lenguaje a utilizar

Después del estudio realizado no se puede decir que existe una plataforma de las analizadas que sea superior, independientemente de que en algunos temas brinden alguna ventaja respecto a la otra, de hecho guardan muchas similitudes en cuanto al campo de acción que cubren y a las facilidades de uso, constituyendo ambas plataformas líderes en la actualidad. Sin embargo la plataforma J2EE reúne un conjunto de elementos que se ajustan mejor a las características de la solución que se pretende plantear para el Módulo de Administración y Control de Recursos, como por ejemplo: es independiente del sistema operativo, es una plataforma madura y con una amplia gama de proveedores, documentación, etc. lo que da más posibilidades desde el punto de vista de poder elegir entre diferentes productos y herramientas, las que más se ajusten al problema en cuestión.

Además de las ventajas que brinda desde el punto de vista tecnológico se cuenta con limitaciones legales por parte del cliente: el decreto Ley No. 3.390 de la Gaceta Oficial de la República Bolivariana de Venezuela establece que la Administración Pública Nacional empleará prioritariamente Software Libre Desarrollado con Estándares Abiertos, en sus Sistemas, Proyectos y Servicios Informáticos, permitiendo el uso de otro tipo de software solo en los casos en los que no se pueda adquirir Software Libre Bajo Estándares Abiertos en tal caso, los órganos y entes de la Administración Pública Nacional deberán solicitar ante el Ministerio de Ciencia y Tecnología, la autorización para adoptar otro tipo de soluciones bajo normas y criterios establecidos por ese Ministerio.

1.13 Conclusiones

En este capítulo se realizó un análisis sobre la atención que se le da en la República Bolivariana de Venezuela a las emergencias que se presentan y el papel de los Centros de Gestión de Emergencias para la solución de estas situaciones.

Se analizó la metodología de desarrollo más idónea para el desarrollo del trabajo, así como el lenguaje de modelado y lenguaje de programación, junto con la plataforma y otras herramientas.

Capítulo II. Descripción de la solución propuesta

2.1 Introducción

En este capítulo se describe la propuesta de solución para el módulo de Administración y Control de Recursos. Se muestra el modelo IDEF, donde se representan los procesos requeridos por el sistema, lo que permite comprender el contexto en que se emplaza el mismo. Se plantean los requisitos funcionales de la aplicación a desarrollar, y se modela la misma teniendo en cuenta los casos de uso y actores que interactúan con ellos. Además se elaboró una descripción detallada para cada caso de uso.

2.2 Modelo del Negocio

A continuación se representan los sujetos que intervienen en la realización de los diferentes procesos identificados.

Tabla 1: Definición de los sujetos.

Actores	Justificación
Especialista de Recursos del Organismo 171.	Representa a las personas encargadas de gestionar los recursos y el mantenimiento de los mismos, o sea al Especialista de Recursos del Organismo y al Administrador de Recursos.
Especialista de Recursos del Organismo.	Representa a la persona en el organismo encargada de gestionar recursos, mantenimientos de recursos y ver las solicitudes de recursos.
Administrador de Recursos.	Representa a la persona en el centro 171 encargada de gestionar organismos y órganos de seguridad, además puede gestionar recursos, solicitudes de recursos y mantenimientos.
Informático.	Representa a la persona que se encarga de gestionar

nuevas categorías y tipos de recursos.

Se realizó el modelado del negocio haciendo uso del método IDEF0. Se muestran los procesos del negocio a través del siguiente diagrama:

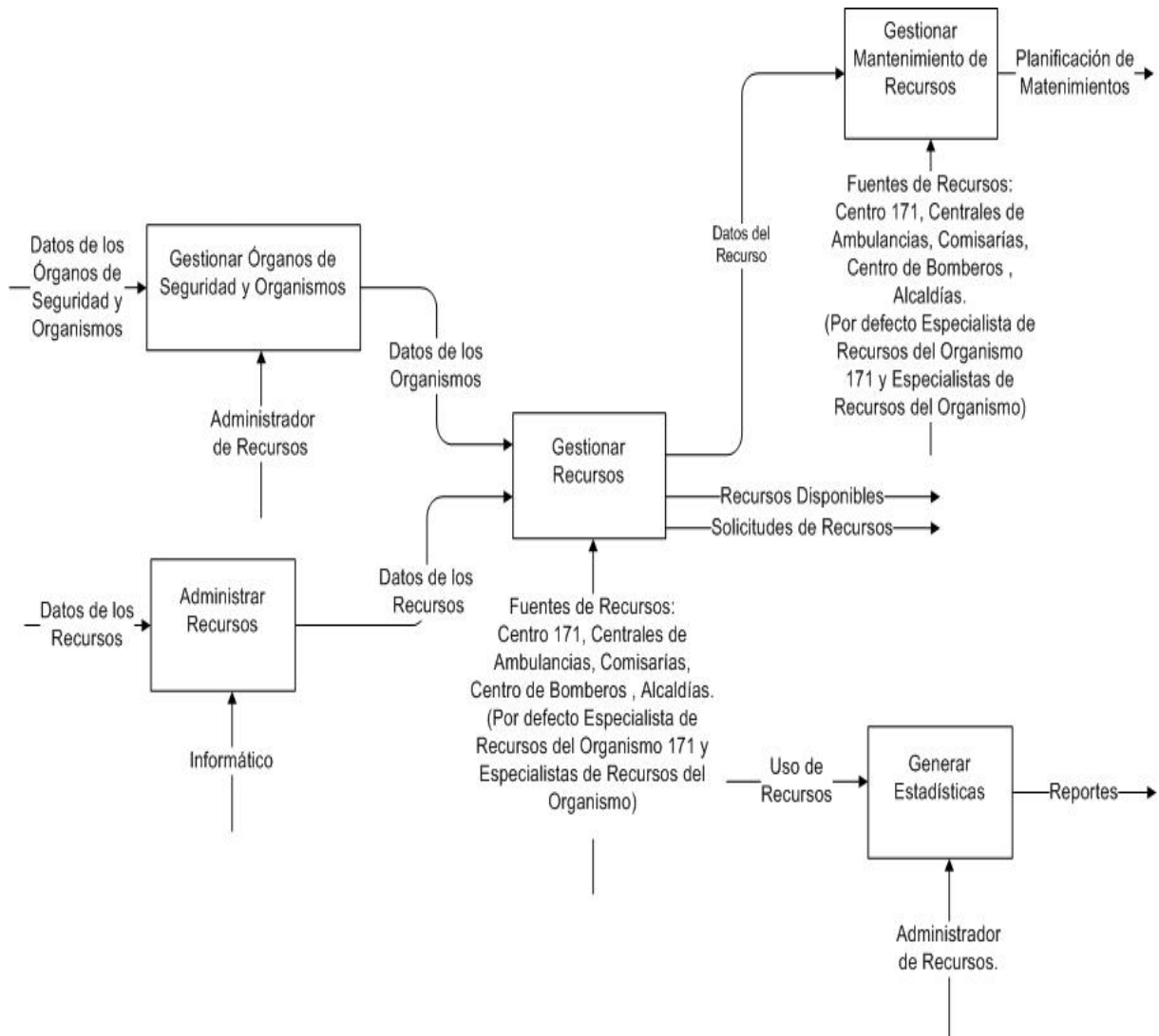


Figura 1: Modelo IDEF.

Descripción de los procesos del negocio:

Gestionar Órganos de Seguridad y Organismos: en este proceso se recogen los datos relacionados a los Órganos de Seguridad y Organismos, los cuales se pueden modificar o eliminar.

Administrar Recursos: consiste en incorporar nuevas categorías y tipos de recursos a manejar, además de los que se consideran como predeterminados, los cuales se pueden eliminar o modificar si es necesario.

Gestionar Recursos: se registran los recursos que disponen los diferentes organismos para la atención de las situaciones que se presenten, así como de los lugares de interés que en algún momento pueden resultar de apoyo. Además se solicitan a los organismos los recursos que se necesitan para fecha y propósito determinado.

Gestionar Mantenimientos de Recursos: se planifican los mantenimientos que se realizarán a los recursos y permite que se pueda acceder a la información referente a los mismos.

Generar Estadísticas: en este proceso se generan estadísticas sobre los recursos y las solicitudes de los mismos y se realizan reportes con sus datos.

2.3 Especificación de los requisitos de software

2.4.1 Requerimientos Funcionales

R1: Autenticar usuario.

- 1.1 La información a solicitar es: nombre y 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 las opciones del menú de acuerdo al usuario autenticado.
- 2.2 Los usuarios pueden ser: Especialista de Recursos del Organismo, Habilitador, Informático y Administrador de Recursos.
 - 2.2.1 Para el usuario Especialista de Recurso:

2.2.2.1 Mostrar las solicitudes de recursos del organismo al que este pertenece. De cada solicitud de recurso:

2.2.2.2.1 Fecha en que se necesitan los recursos.

2.2.2.2.2 Asunto de la solicitud.

2.2.2.2.3 Por ciento de satisfacción de cada solicitud.

R3: Mostrar mantenimiento según configuración.

3.1 Buscar datos de los mantenimientos planificados a los recursos del organismo autenticado.

3.1.1 Criterio de búsqueda:

3.1.1.1 Rango de tiempo de búsqueda definido en la configuración.

3.1.2 Mostrar el resultado de la búsqueda:

3.1.2.1 Fecha de realización del mantenimiento.

3.1.2.2 Nombre del tipo de recurso.

3.1.2.3 Código del recurso.

R4: Registrar órgano de seguridad.

4.1 Insertar los datos de un nuevo órgano de seguridad:

4.1.1 Nombre.

R5: Mostrar listado de órganos de seguridad.

5.1 Buscar todos los órganos de seguridad registrados.

5.2 Mostrar el listado de nombres de los órganos de seguridad obtenidos.

R6: Modificar órgano de seguridad.

6.1 Insertar el nuevo nombre del órgano de seguridad seleccionado.

R7: Eliminar órgano de seguridad.

7.1 Seleccionar el órgano de seguridad que se desea eliminar.

7.2 Marcar el órgano de seguridad como eliminado en la base de datos.

7.3 Actualizar la lista de órganos de seguridad mostrada.

R8: Registrar organismo.

8.1 Insertar los datos referentes al nuevo organismo.

8.1.1 Órgano de seguridad.

8.1.2 Nombre del organismo.

8.1.3 Dirección: municipio, parroquia, avenida, calle, caserío, aldea, ubicación, piso, teléfono, fotos.

8.1.4 Recursos a gestionar:

8.2.4.1 Tipo de Recurso.

R9: Mostrar listado de organismos.

9.1 Buscar datos de organismos.

9.1.1 Criterio de búsqueda:

9.1.1.1 Código.

9.1.1.2 Nombre.

9.1.1.3 Teléfono.

9.1.2 Mostrar el resultado de la búsqueda:

9.1.2.1 Nombre.

9.1.2.2 Teléfono.

R10: Ver detalles de un organismo.

10.1 Buscar datos de detalle de un organismo.

10.1.1 Criterio de búsqueda:

10.1.1.1 Código.

10.1.2 Mostrar el resultado de la búsqueda:

10.1.2.1 Nombre.

10.1.2.2 Órgano de seguridad.

10.1.2.3 Dirección: municipio, parroquia, avenida, calle, caserío, aldea, ubicación, piso, teléfono, fotos.

10.1.2.4 Teléfono.

10.1.2.5 Lista de fotos.

10.1.2.6 Lista de recursos que gestiona.

R11: Modificar datos de un organismo.

11.1 Modificar los datos del organismo seleccionado.

11.1.1 Los datos que se pueden modificar son:

11.1.1.1 Órgano de seguridad.

11.1.1.2 Nombre del organismo.

11.1.1.3 Dirección: municipio, parroquia, avenida, calle, caserío, aldea, ubicación, piso, teléfono, fotos.

11.1.1.4 Listado de recursos a gestionar:

11.1.1.5 Tipo de Recurso.

11.1.1.6 Listado de órganos de seguridad asociados al organismo.

11.1.1.7 Listado de tipos de recursos asociados al organismo.

R12: Eliminar organismo.

12.1 Marcar el organismo seleccionado como eliminado en la base de datos.

12.2 Actualizar la lista de organismos mostrada.

R13: Registrar recurso.

13.1 Obtener categorías y tipos de recurso que existen.

13.2 Construir y mostrar un formulario con los atributos y el tipo de dato requeridos para el recurso que se desea insertar.

13.3 Verificar que no existe en la base de datos un recurso con el atributo identificador igual al introducido.

R14: Mostrar listado de recursos.

14.1 Buscar los datos de un recurso.

14.1.1 Criterio de Búsqueda:

14.1.1.1 Atributos del tipo de recurso seleccionado definidos como criterios de búsqueda.

14.1.1.2 Mostrar un campo para cada criterio de búsqueda teniendo en cuenta el tipo de dato.

14.1.2 Mostrar el resultado de la búsqueda.

14.1.2.1 Categoría.

14.1.2.2 Tipo de recurso.

14.1.2.3 Atributo que identifica al recurso según la categoría:

14.1.2.3.1 Si la categoría es "Móvil" el atributo es matrícula.

14.1.2.3.2 Si la categoría es "Fijo" los atributos varían en función del tipo de recurso.

14.1.2.3.3 Si la categoría es "Accesorio" el atributo es número de serie.

R15: Ver detalles de recursos.

- 15.1 Obtener los atributos del tipo de recurso al que pertenece el recurso seleccionado y el tipo de dato de cada atributo.
- 15.2 Crear un formulario con los atributos obtenidos y el tipo de dato de cada atributo.
- 15.3 Buscar datos de detalle del recurso.
 - 15.3.1 Criterio de Búsqueda:
 - 15.3.1.1 Código del recurso.
 - 15.3.2 Mostrar formulario creado con el resultado de la búsqueda.
 - 15.3.2.1 Atributos del recurso según el tipo de recurso.

R16: Modificar recursos.

- 16.1 Obtener de la base de datos los atributos del tipo de recurso al que pertenece el recurso seleccionado y el tipo de dato de cada atributo.
- 16.2 Crear un formulario con los atributos obtenidos y el tipo de dato de cada atributo.
- 16.3 Buscar los datos del recurso seleccionado según el tipo de recurso.
- 16.4 Modificar los datos del recurso seleccionado.

R17: Eliminar recursos.

- 17.1 Marcar en la base de datos como eliminado los datos del recurso seleccionado.
- 17.2 Actualizar la lista de recursos mostrada.

R18: Asociar accesorios a recursos.

- 18.1 Obtener el tipo de recurso al que pertenece el recurso seleccionado y el atributo que lo identifica.
- 18.2 Obtener todos los accesorios cuyo número de serie no está asociado a ningún recurso.
- 18.3 Crear un formulario con los atributos obtenidos.
- 18.4 Asociar los números de serie seleccionados al recurso.

R19: Solicitar recursos.

- 19.1 Insertar datos referentes a la solicitud:
 - 19.1.1 Asunto.
 - 19.1.2 Descripción.
 - 19.1.3 Fecha en que se necesitan los recursos.
 - 19.1.4 Órgano de seguridad.

19.1.5 Organismo.

19.1.6 Tipo de recurso

19.1.7 Cantidad

19.1.8 Mostrar un listado de solicitudes con el tipo de recurso y la cantidad de recursos solicitados.

R20: Mostrar listado de solicitudes de recurso.

20.1 Buscar datos de solicitudes de recursos.

20.1.1 Criterio de Búsqueda:

20.1.1.1 Fecha en que se necesitan los recursos.

20.1.1.2 Fecha en que se realizó la petición.

20.1.2 Calcular el por ciento de satisfacción de cada solicitud de recursos.

20.1.3 Mostrar el resultado de la búsqueda.

20.1.3.1 Asunto.

20.1.3.2 Fecha en que se necesitan los recursos.

20.1.3.3 Por ciento de satisfacción.

R21: Ver detalles de una solicitud de recursos.

21.1 Buscar datos de detalle de una solicitud de recurso:

21.1.1 Criterio de Búsqueda:

21.1.1.1 Código de la solicitud de recurso.

21.1.2 Mostrar el resultado de la búsqueda.

21.1.2.1 Asunto.

21.1.2.2 Descripción.

21.1.2.3 Fecha en que se necesitan los recursos.

21.1.2.4 Por ciento de satisfacción de la solicitud.

21.1.2.5 Lista de recursos solicitados:

21.1.2.5.1 Tipo de recurso.

21.1.2.5.2 Cantidad solicitada.

21.1.2.5.3 Cantidad de recursos con el estado Habilitado que tiene el organismo.

R22: Modificar el estado de un recurso.

- 22.1 Cambiar el estado de un recurso.
 - 22.1.1 Puede ser Habilitado o Deshabilitado.

R23: Crear categoría.

- 23.1 Insertar datos de una nueva categoría.
 - 23.1.1 Nombre.

R24: Modificar categoría.

- 24.1 Insertar el nuevo nombre de la categoría seleccionada.
 - 24.1.1 Solo se podrán modificar aquellas categorías que no son predeterminadas, es decir las que hayan sido insertadas por el informático.

R25: Crear tipo de recurso.

- 25.1 Insertar datos del nuevo tipo de recurso.
 - 25.1.1 Nombre de la entidad.
 - 25.1.2 Categoría.
 - 25.1.3 Especificar si tiene mantenimiento, fotos y planos.
 - 25.1.4 Recursos con los que tiene relación.
 - 25.1.5 Atributos.
 - 25.1.5.1 Nombre.
 - 25.1.5.2 Tipo de dato.
 - 25.1.5.3 Tamaño.
 - 25.1.5.4 Especificar si es nulo.
 - 25.1.5.5 Especificar identificador.
 - 25.1.5.6 Especificar si va a ser criterio de búsqueda.
- 25.2 Crear una nueva tabla en la base de datos para el tipo de recurso.

R26: Mostrar datos de un tipo de recurso.

- 26.1 Buscar datos de tipos de recurso.
 - 26.1.1 Criterio de búsqueda:
 - 26.1.1.1 Categoría de recurso.
 - 26.1.1.2 Tipo de recurso.
 - 26.1.2 Mostrar el resultado de la búsqueda.

- 26.1.2.1 Mantenimiento.
- 26.1.2.2 Fotos.
- 26.1.2.3 Planos.
- 26.1.2.4 Listado de recursos asociados.
- 26.1.2.5 De los atributos de dicho tipo de recurso mostrar:
 - 26.1.2.5.1 Nombre.
 - 26.1.2.5.2 Tipo de Dato.
 - 26.1.2.5.3 Tamaño.
 - 26.1.2.5.4 Si es nulo.
 - 26.1.2.5.5 Identificador.
 - 26.1.2.5.6 Criterio de búsqueda.

R27: Modificar tipo de recurso.

- 27.1 Mostrar los datos del tipo de recurso seleccionado y modificar los que se desean.
 - 27.1.1 Sólo se pueden modificar los atributos de los tipos de recurso que no sean predeterminados en la plantilla del tipo de recurso.

R28: Eliminar tipo de recurso.

- 28.1 Verificar que el tipo de recurso seleccionado se pueda eliminar.
 - 28.1.1 Se puede eliminar si el tipo de recurso no es predeterminado.
- 28.2 Marcar como eliminado de la base de datos el tipo de recurso.

R29: Generar expediente de recursos.

- 29.1 Obtener los datos del expediente del recurso.
 - 29.1.1 Criterio de búsqueda:
 - 29.1.1.1 Código del recurso.
- 29.2 Mostrar datos del expediente de recurso.
 - 29.2.1 Fecha.
 - 29.2.2 Todos los atributos registrados en la base de datos correspondientes al recurso seleccionado. Estos varían en dependencia de la categoría y el tipo de recurso.
 - 29.2.3 Listado de recursos asociados.
 - 29.2.4 Nombre.
 - 29.2.5 Identificador.

29.2.6 Mantenimientos.

29.2.6.1 Fecha del próximo mantenimiento.

29.2.6.2 Últimos mantenimientos realizados.

29.2.6.2.1 Fecha.

29.2.6.2.2 Descripción.

29.2.6.3 Incidencias.

29.2.6.3.1 Fecha.

R30: Generar reportes de solicitudes de recursos.

30.1 Mostrar un formulario que permite seleccionar estadísticas de cumplimiento de las solicitudes de recursos por períodos:

30.1.1 Semanal.

30.1.2 Mensual.

30.1.3 Trimestral.

30.1.4 Semestral.

30.1.5 Anual.

30.2 Generar reporte de solicitudes de recursos:

30.2.1 Generales.

11.2.1.1 Organismo.

11.2.1.2 Fecha.

11.2.1.3 Por Ciento de Satisfacción.

30.2.2 Recursos más solicitados.

30.2.2.1 Tipo de recurso.

30.2.3 Organismos con mayor por ciento de satisfacción.

30.2.3.1 Nombre del organismo.

30.2.4 Organismos con menor por ciento de satisfacción.

30.2.4.1 Nombre del organismo.

30.2.5 Promedios.

30.2.5.1 Tiempo de respuesta promedio de los organismos.

30.2.5.2 Por ciento de satisfacción promedio.

30.2.5.3 Promedio de cantidad de recursos por solicitud.

R31 Planificar mantenimiento de recurso.

- 31.1 Obtener el tipo de recurso y el atributo que identifica al recurso seleccionado.
- 31.2 Crear un formulario teniendo en cuenta el tipo de recurso, para introducir los datos referentes al mantenimiento.
 - 31.1.1 Fecha del mantenimiento.
 - 31.1.2 Periodicidad.

R32: Registrar mantenimiento de recurso.

- 32.1 Obtener categorías y tipos de recurso que existen.
- 32.2 Introducir los datos:
 - 32.2.1 Atributo que identifica al tipo de recurso.
 - 32.2.2 Fecha en que se necesitan los recursos.
 - 32.2.3 Descripción.

R33: Mostrar mantenimientos.

- 33.1 Buscar mantenimientos de recursos.
 - 33.1.1 Criterios de búsqueda:
 - 33.1.1.1 Fecha de inicio.
 - 33.1.1.2 Fecha de fin.
 - 33.1.2 Mostrar el resultado de la búsqueda:
 - 33.1.2.1 Fecha de mantenimiento.
 - 33.1.2.2 Tipo de recurso.
 - 33.1.2.3 Código del recurso.

R34: Ver detalles de mantenimiento.

- 34.1 Mostrar datos de mantenimiento del recurso:
 - 34.1.1 Tipo de recurso.
 - 34.1.2 Atributo que lo identifica.
 - 34.1.3 Varía en dependencia del tipo de recurso.
 - 34.1.4 Fecha del próximo mantenimiento.
 - 34.1.5 Periodicidad.

2.4 Modelo del Sistema

2.4.1 Actores y Casos de Uso del sistema

Tabla 2: Definición de los actores del sistema a automatizar.

Actores	Justificación
Usuario.	Representa a todos los usuarios que tienen acceso a la aplicación, o sea al Especialista de Recursos del Organismo 171, al Habilitador y al Informático.
Especialista de Recursos del Organismo 171.	Representa a las personas encargadas de gestionar los recursos y el mantenimiento de los mismos, o sea al Especialista de Recursos del Organismo y al Administrador de Recursos.
Especialista de Recursos del Organismo.	Representa a la persona en el organismo encargada de gestionar recursos, mantenimientos de recursos y ver las solicitudes de recursos.
Administrador de Recursos.	Representa a la persona en el centro 171 encargada de gestionar organismos y órganos de seguridad, además puede gestionar recursos, solicitudes de recursos y mantenimientos.
Habilitador.	Representa a la persona encargada de habilitar o deshabilitar los recursos en el sistema para que puedan ser usados o no en la atención de las emergencias.
Informático.	Representa a la persona que se encarga de gestionar nuevas categorías y tipos de recursos.

A continuación se presentan los Casos de Uso (CU) identificados para satisfacer los requerimientos funcionales del sistema:

2.4.2 Diagrama de Casos de Uso del Sistema

Para una mejor organización se agruparon los casos de uso por paquetes teniendo en cuenta su propósito. A continuación se muestra el diagrama de paquetes:

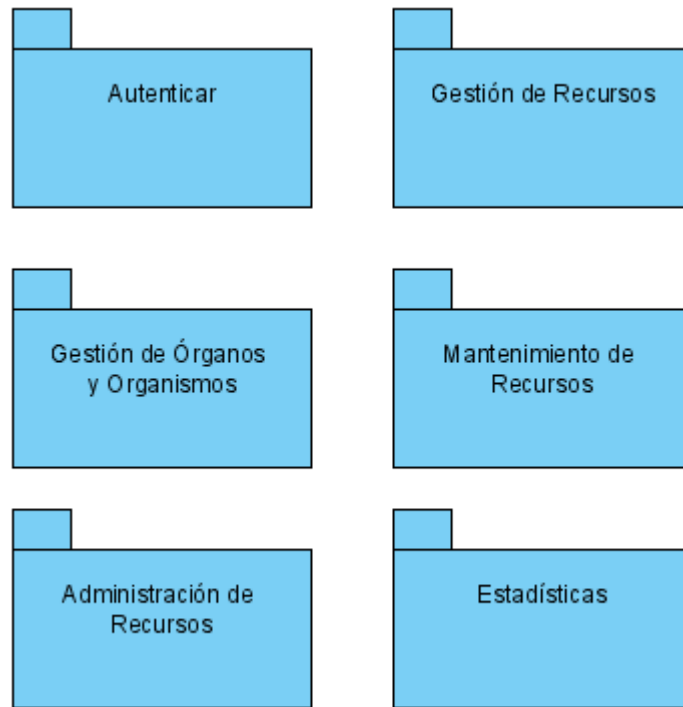


Figura 2: Diagrama de Paquetes del módulo de Administración y Control de Recursos.

A continuación se representan los diagramas de casos de uso por paquetes:

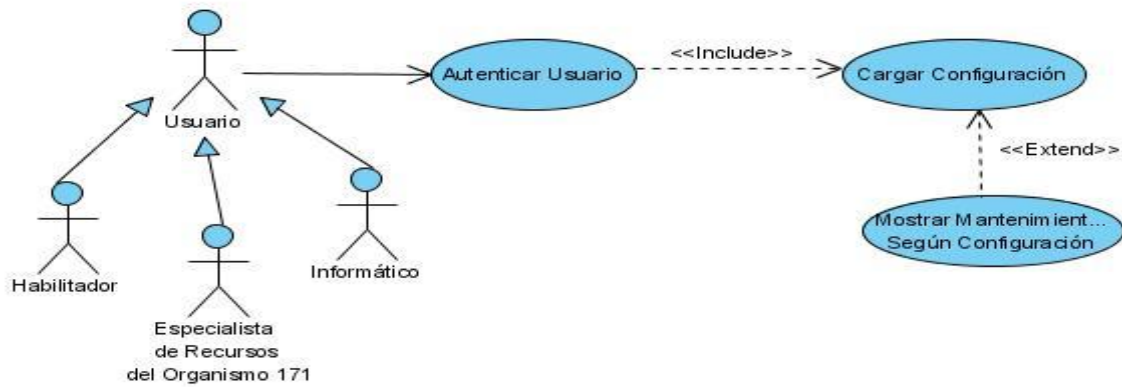


Figura 3: Diagrama de Casos de Uso del Paquete Autenticación.

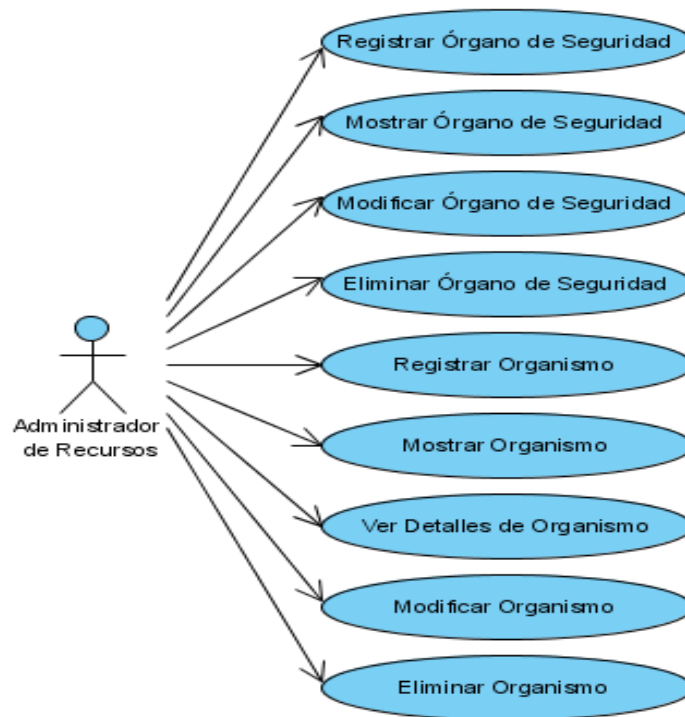


Figura 4: Diagrama de Casos de Uso del Paquete Gestión de Órganos y Organismos.

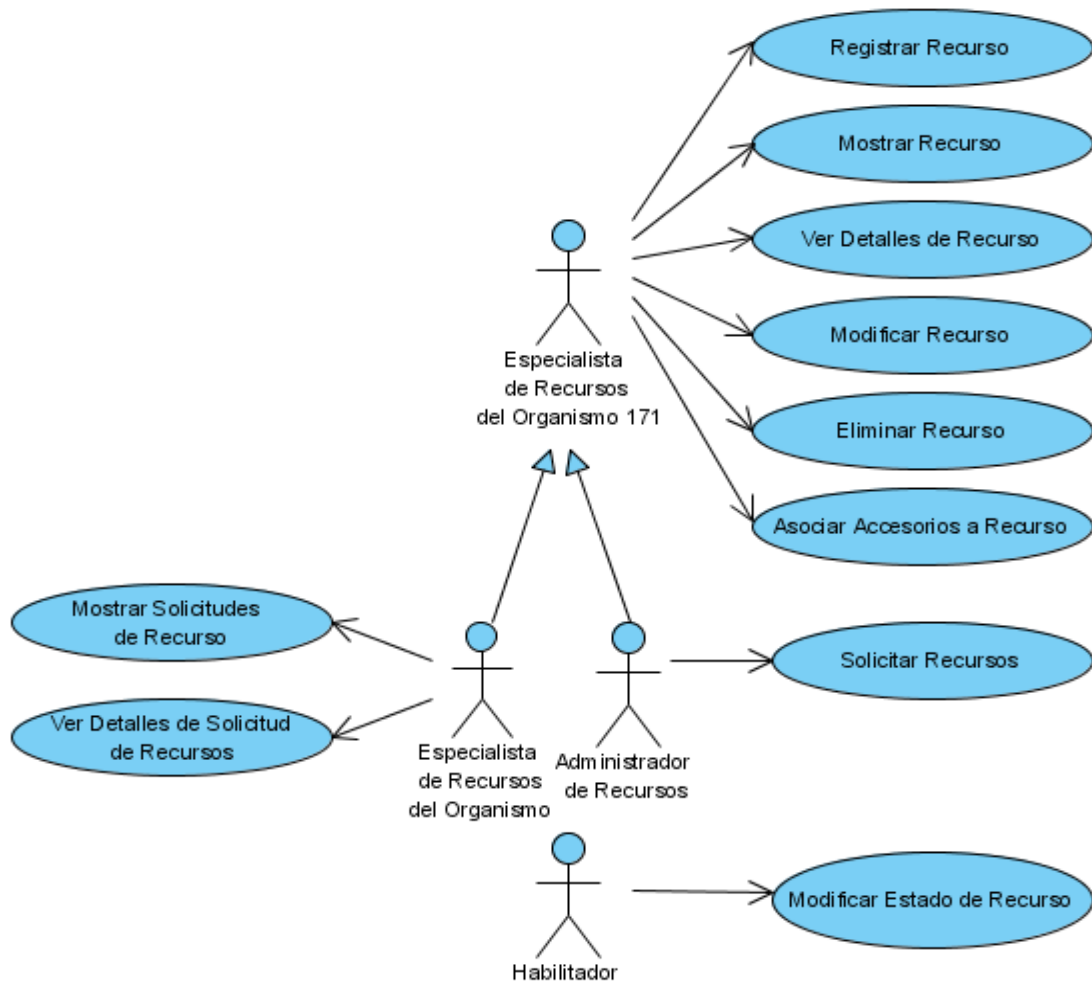


Figura 5: Diagrama de Casos de Uso del Paquete Gestión de Recursos.

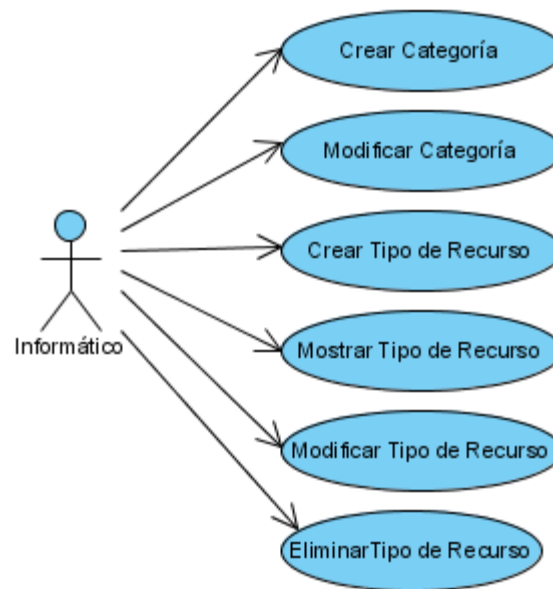


Figura 6: Diagrama de Casos de Uso del Paquete Administración de Recursos.

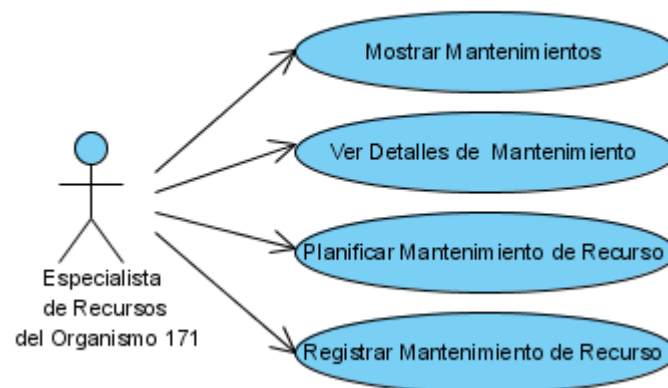


Figura 7: Diagrama de Casos de Uso del Paquete Mantenimiento de Recursos.

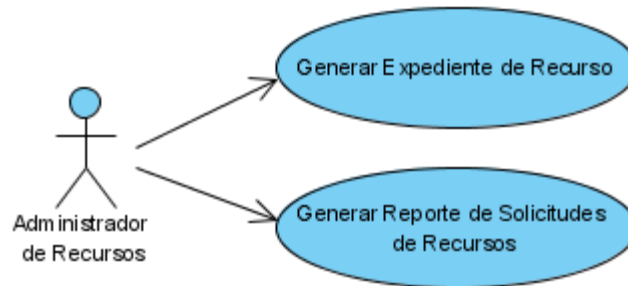


Figura 8: Diagrama de Casos de Uso del Paquete Estadísticas.

Basado en las características de RUP, se hizo una división de los CU en 2 ciclos de desarrollo. A continuación se muestra una tabla con el nombre de cada caso de uso, el ciclo al que corresponde y la justificación de la selección del ciclo.

Tabla 3: Casos de Uso por ciclos.

Nombre de Caso de Uso	Ciclo	Justificación de la selección
Autenticar Usuario.	1	Estos Casos de Uso se encuentran implicados en el primer ciclo porque en su conjunto conforman las funcionalidades básicas del Sistema de Administración y Control de Recursos y son necesarios para el funcionamiento del centro 171.
Cargar Configuración.		
Registrar Órgano de Seguridad.		
Mostrar Órgano de Seguridad.		
Modificar Órgano de Seguridad.		
Eliminar Órgano de Seguridad.		
Registrar Organismo.		
Mostrar Organismo.		
Ver Detalles de Organismo.		
Modificar Organismo.		
Eliminar Organismo.		
Registrar Recurso.		
Asociar Accesorios a Recursos.		
Modificar Recurso.		
Modificar Estado de Recurso.		

Eliminar Recurso.		
Mostrar Recurso.		
Ver Detalles de Recurso.		
Solicitar Recursos.		
Mostrar Solicitudes de Recursos.		
Ver Detalles de Solicitud de Recursos.		
Crear Categoría.	2	En este ciclo se implementarán funcionalidades que son importantes para el funcionamiento del sistema que se desea desarrollar pero que no le son imprescindibles a la hora de iniciar su funcionamiento.
Modificar Categoría.		
Crear Tipo de Recurso.		
Modificar Tipo de Recurso.		
Eliminar Tipo de Recurso.		
Mostrar Tipo de Recurso.		
Generar Expediente de Recurso.		
Planificar Mantenimiento de Recurso.		
Mostrar Mantenimientos.		
Ver Detalles de Mantenimiento.		
Mostrar Mantenimientos Según Configuración.		
Registrar Mantenimiento de Recurso.		
Generar Reportes de Solicitudes de Recursos.		

2.4.3 Descripción textual de los Casos de Uso

A continuación se muestran las descripciones detalladas de los casos que se desarrollarán en la primera iteración.

CU Autenticar Usuario.

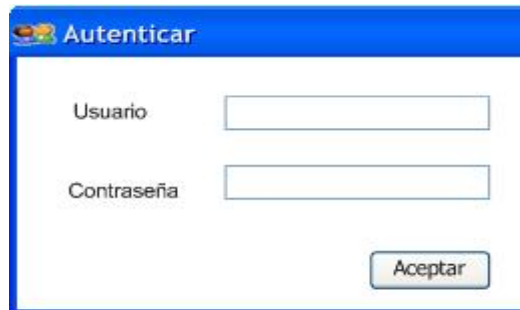
Propósito.

Permitir que el usuario inicie sesión en la aplicación.

Descripción.

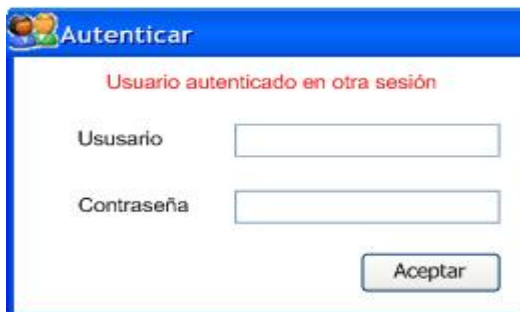
El usuario introduce los datos para autenticarse, se verifica que los datos introducidos son correctos y que el usuario cuenta con los permisos necesarios para autenticarse en el sistema. En caso contrario no se permite la autenticación del usuario. También se determina si el usuario está autenticado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

Diseño de Interfaz de Usuario.



The screenshot shows a window titled 'Autenticar' with a blue header bar. Inside the window, there are two text input fields: the first is labeled 'Usuario' and the second is labeled 'Contraseña'. Below these fields is a button labeled 'Aceptar'.

Figura 9 Autenticar Usuario.



The screenshot shows the same 'Autenticar' window as in Figure 9. At the top of the main content area, there is a red text message: 'Usuario autenticado en otra sesión'. Below this message are the 'Usuario' and 'Contraseña' input fields, and the 'Aceptar' button.

Figura 10. Usuario Autenticado en otra sesión.

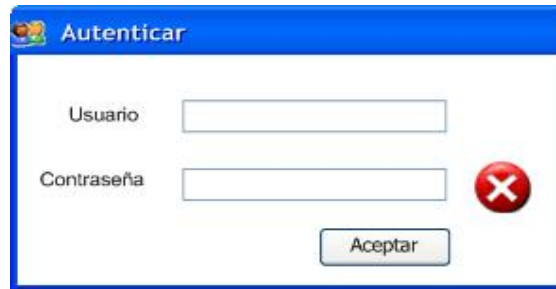


Figura 11. Contraseña Incorrecta.

Precondiciones.

No aplica.

Tabla 4: Flujo básico del CU Autenticar Usuario.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El usuario introduce el 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 ese usuario.
	6. Obtiene de la BD los siguientes datos del usuario autenticado: <ul style="list-style-type: none"> • Código del usuario. • Código del Organismo al que pertenece.

	7. Comprueba en la BD que no exista otra sesión abierta para ese usuario en la aplicación usando el 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 Alterno 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 ese usuario en la aplicación.
	7a.2. Muestra un mensaje indicando que el usuario ya está autenticado en la aplicación.
Referencias.	R1

CU Cargar Configuración

Propósito.

Cargar los elementos de configuración definidos para el usuario autenticado.

Descripción.

Obtiene de la BD las opciones a las que tiene acceso el usuario autenticado.

Si el usuario es un especialista de recursos del organismo muestra las solicitudes de recursos y los mantenimientos planificados para los recursos del organismo. Obtiene las solicitudes de recursos pertenecientes al organismo cuya fecha en que se necesitan los recursos sea mayor o igual que la actual usando el código del organismo. Se calcula el por ciento de satisfacción para cada una de las solicitudes.

Obtiene de la BD los mantenimientos de recursos pertenecientes al organismo cuya fecha de planificación del mantenimiento se encuentre dentro del periodo de tiempo (definido en la configuración del sistema), que se usará para la visualización de estos mantenimientos (por ejemplo, semana actual, mes actual,

etc.). Muestra la página principal con las opciones posibles al usuario autenticado, el listado de mantenimientos de recursos y lista de solicitudes de recursos ordenada por fecha en que se necesitan los recursos.

Diseño de Interfaz de Usuario.

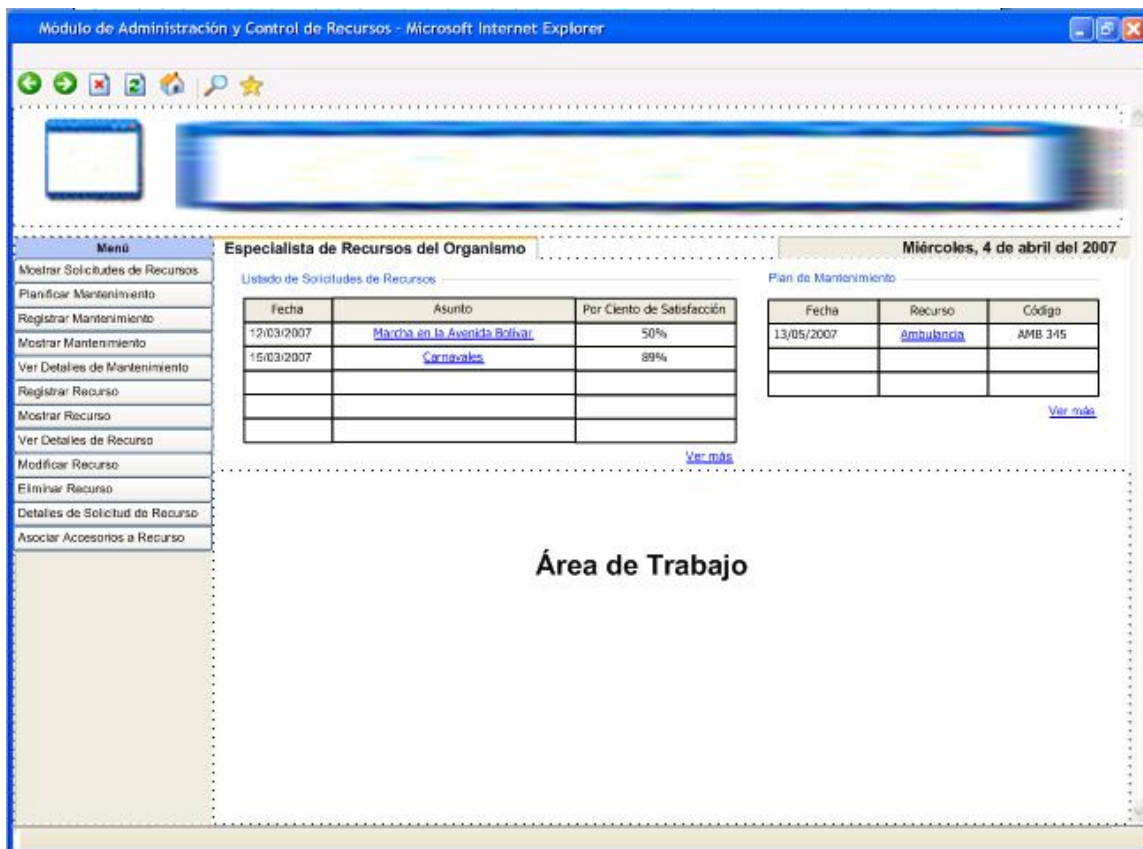


Figura 12. Página Principal (Ejemplo: Especialista de Recursos del Organismo).

Precondiciones.

No aplica.

Tabla 5: Flujo básico del CU Cargar Configuración.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
	1. Comprueba en la BD el rol del usuario autenticado.

	<ul style="list-style-type: none"> • Si el usuario autenticado es Especialista de Recursos del Organismo ir a la sección “Configuración del Especialista de Recursos del Organismo”. • Si el usuario autenticado es Administrador de Recursos ir a la sección “Configuración del Administrador de Recursos”. • Si el usuario autenticado es Informático ir a la sección “Configuración del Informático”. • Si el usuario autenticado es Habilitador ir a la sección “Configuración del Habilitador”.
<p>Sección “Configuración del Especialista de Recursos del Organismo”</p>	
	<p>2. Habilita las opciones:</p> <ul style="list-style-type: none"> • “Mostrar Solicitudes de Recursos”. • “Mostrar Mantenimiento”. • “Planificar Mantenimiento”. • “Registrar Mantenimiento”. • “Ver Detalles de Mantenimiento”. • “Ver Detalles de Solicitud de Recurso”. • “Registrar Recurso”. • “Mostrar Recurso”. • “Ver Detalles de Recurso”. • “Modificar Recurso”. • “Eliminar Recurso”. • “Asociar Accesorios a Recurso”.
	<p>3. Obtiene de la BD el código del organismo correspondiente al usuario autenticado.</p>
	<p>4. Comprueba que existen solicitudes de recursos para el organismo del usuario</p>

	autenticado.
	5. Obtiene de la BD las solicitudes de recursos donde el código del organismo coincida con el código del organismo del usuario autenticado y la fecha en que se necesitan los recursos sea mayor o igual que la fecha actual.
	6. Calcula el por ciento de satisfacción para cada una de las solicitudes. El por ciento se obtiene usando la cantidad total de recursos solicitados y la cantidad total de recursos otorgados y se calcula qué por ciento representa la asignación con respecto a lo pedido.
	7. Muestra las solicitudes de recursos, con los campos: <ul style="list-style-type: none"> • Fecha en que se necesitan los recursos. • Asunto de la solicitud. • Por ciento de satisfacción.
	8. Habilita la opción “Ver más” para ver los detalles de una solicitud de recursos.
	9. Invoca al CU Mostrar Mantenimiento Según Configuración usando el código del usuario autenticado.
Flujo Alternativo 4a No Existen Solicitudes de Recursos	
Acción del Actor	Respuesta del Sistema
	4a.1. Comprueba que no existen solicitudes de recursos para el organismo del usuario autenticado. Ir a la acción 9.

Sección "Configuración del Administrador de Recursos"	
	<p>2. Habilita las opciones:</p> <ul style="list-style-type: none"> • "Mostrar Mantenimientos". • "Planificar Mantenimiento". • "Registrar Mantenimiento". • "Ver Detalles de Mantenimiento". • "Registrar Recurso". • "Mostrar Recurso". • "Ver Detalles de Recurso". • "Modificar Recurso". • "Eliminar Recurso". • "Registrar Órgano de Seguridad". • "Mostrar Órganos de Seguridad". • "Modificar Órganos de Seguridad". • "Eliminar Órganos de Seguridad". • "Registrar Organismo". • "Mostrar Organismo". • "Ver Detalles de Organismo". • "Modificar Organismo". • "Eliminar Organismo". • "Generar Expediente de Recurso". • "Generar Reporte de Solicitud de Recursos". • "Solicitar Recursos". • "Asociar Accesorios a Recurso".
	<p>3. Invoca el CU Mostrar Mantenimientos Según Configuración usando el código del usuario autenticado.</p>
Sección "Configuración del Informático"	

	<p>2. Habilita las opciones:</p> <ul style="list-style-type: none"> • “Crear Categoría”. • “Modificar Categoría”. • “Crear Tipo de Recurso”. • “Mostrar Tipo de Recurso”. • “Modificar Tipo de Recurso”. • “Eliminar Tipo de Recurso”.
Sección “Configuración del Habilitador”	
	<p>2. Habilita la opción:</p> <ul style="list-style-type: none"> • “Modificar Estado de Recurso”.
Referencias.	R2

CU Registrar Órgano de Seguridad.

Propósito.

Registrar los órganos de seguridad en el sistema.

Descripción.

El Administrador de Recursos selecciona la opción “Registrar Órgano de Seguridad”. Se muestra un formulario con los datos que se deben insertar. El Administrador de Recursos introduce los datos especificados. Se almacenan los datos en la BD.

Diseño de Interfaz de Usuario.



Figura 13. Registrar Órgano de Seguridad.

Precondiciones.

No aplica.

Tabla 6: Flujo básico del CU Registrar Órgano de Seguridad.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1.El Administrador de Recursos selecciona la opción “Registrar Órgano de Seguridad”	2. Obtiene de la BD el nombre de los órganos de seguridad que existen registrados.
	3. Muestra el formulario Registrar Órgano de Seguridad.
4. El Administrador de Recursos introduce el nombre del órgano de seguridad.	
5. El Administrador de Recursos selecciona la opción “Aceptar”.	6. Comprueba que el dato sea correcto.
	7. Comprueba que no existe ningún órgano en

	la BD usando el nombre introducido.
	8. Almacena la información en la BD.
	9. Muestra el nombre del órgano de seguridad registrado, en la lista.
Flujo Alterno 3*a Opción "Cancelar".	
Acción del Actor	Respuesta del Sistema
3*a.1. El Administrador de Recursos selecciona la opción "Cancelar".	6a.1. Retorna a la página anterior.
Flujo Alterno 6a Dato Introducido Incorrecto.	
Acción del Actor	Respuesta del Sistema
	6a.1. Comprueba que el dato es incorrecto.
	6a.2. Indica que el dato está incorrecto.
Flujo Alterno 7a Órgano Registrado Anteriormente.	
Acción del Actor	Respuesta del Sistema
	7a.1. Comprueba que existe ese órgano en la BD usando el nombre introducido.
	7a.2. Muestra un mensaje indicando que el órgano de seguridad ya se encuentra almacenado.
Referencias.	R12

CU Mostrar Órgano de Seguridad.

Propósito.

Mostrar el listado de los órganos de seguridad con los cuales interactúa el Centro 171.

Descripción.

Obtiene de la BD la información asociada a los órganos de seguridad y la muestra.

Diseño de Interfaz de Usuario.

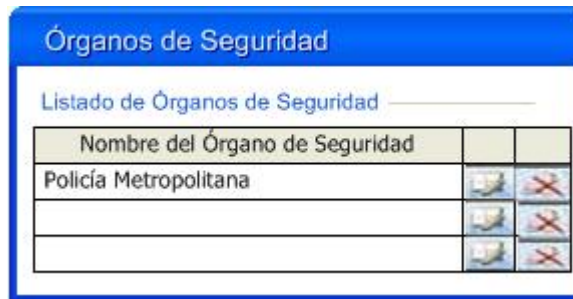


Figura 14. Mostrar Órganos de Seguridad.

Precondiciones.

No aplica.

Tabla 7: Flujo básico del CU Mostrar Órgano de Seguridad.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona la opción "Mostrar Órgano de Seguridad"	2. Obtiene de la BD los órganos de seguridad registrados.
	3. Comprueba que existen órganos de seguridad registrados.
	4. Muestra el listado de los órganos de seguridad obtenidos con el siguiente dato: <ul style="list-style-type: none"> • Nombre del órgano de seguridad.
	5. Habilita las opciones: <ul style="list-style-type: none"> • "Modificar". • "Eliminar".
Flujo Alternativo 3a No Existen Órganos de Seguridad	
Acción del Actor	Respuesta del Sistema
	3a.1. Comprueba que no existen órganos de seguridad registrados.

	3a.2. Muestra un mensaje indicando que no existen órganos de seguridad registrados.
Referencias.	R13

CU Modificar Órgano de Seguridad.

Propósito.

Modificar los datos asociados a un órgano de seguridad.

Descripción.

El Administrador de Recursos selecciona el elemento que desea modificar. Usando el código del órgano de seguridad seleccionado se muestra un formulario con los datos asociados al órgano de seguridad de forma editable. El Administrador de Recursos modifica los datos. Se actualizan los datos en la BD.

Diseño de Interfaz de Usuario.

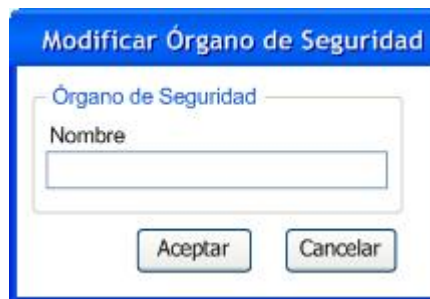


Figura 15. Modificar Órgano de Seguridad.

Precondiciones.

Debe estar mostrada la lista de órganos de seguridad.

Tabla 8: Flujo básico del CU Modificar Órgano de Seguridad.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona un órgano de seguridad de la lista mostrada.	

2. El Administrador de Recursos selecciona la opción "Modificar".	3. Obtiene de la BD, usando el código del órgano de seguridad seleccionado, los datos: <ul style="list-style-type: none"> • Código. • Nombre.
	4. Muestra el formulario Modificar Órgano de Seguridad.
5. El Administrador de Recursos modifica el nombre del órgano de seguridad.	
6. El Administrador de Recursos selecciona la opción "Aceptar".	7. Comprueba que el dato es correcto.
	8. Actualiza en la BD, usando el código del órgano de seguridad seleccionado, el nombre del mismo.
Flujo Alterno 4*a Cancelar	
Acción del Actor	Respuesta del Sistema
4*a.1. El Administrador de Recursos selecciona la opción "Cancelar".	4*a.2. Retorna a la página anterior.
Flujo Alterno 7a Dato Introducido Incorrecto	
Acción del Actor	Respuesta del Sistema
	7a.1. Comprueba que el dato es incorrecto.
	7a.2. Indica que el dato está incorrecto.
Referencias.	R14

CU Eliminar Órgano de Seguridad.

Propósito.

Eliminar un órgano de seguridad del sistema.

Descripción.

El Administrador de Recursos selecciona uno o varios órganos de seguridad para eliminar. Se le muestra un mensaje de confirmación de eliminación. El Administrador de Recursos acepta la eliminación. Se eliminan de la BD los órganos de seguridad seleccionados.

Diseño de Interfaz de Usuario.

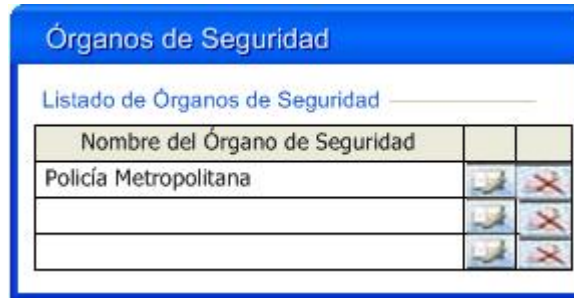


Figura 16. Mostrar Órgano de Seguridad.

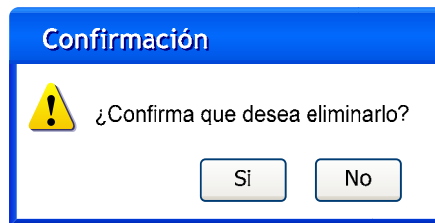


Figura 17. Confirmación de Eliminación de Órgano de Seguridad.

Precondiciones.

Debe estar mostrada la lista de órganos de seguridad.

Tabla 9: Flujo básico del CU Eliminar Órgano de Seguridad.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona uno o varios órganos de seguridad.	
2. El Administrador de Recursos selecciona la opción "Eliminar".	3. Muestra un mensaje de confirmación de eliminación.
4. El Administrador de Recursos	5. Marca como eliminado en la BD los órganos

selecciona la opción "Si".	de seguridad seleccionados.
	6. Marca como eliminado los organismos asociados al o los órganos de seguridad seleccionados.
	7. Actualiza la lista de órganos de seguridad mostrada.
Flujo Alterno 4a Opción "No".	
Acción del Actor	Respuesta del Sistema
4a.1. El Administrador de Recursos selecciona la opción "No".	4a.2. Cierra el mensaje mostrado.
Referencias.	R15

CU Registrar Organismo.

Propósito.

Registrar los organismos en el sistema.

Descripción.

El sistema permite definir de qué Órgano es el Organismo a insertar. Se muestra un formulario con los datos que se deben insertar. El Administrador de Recursos introduce los datos especificados. Se almacenan los datos en la BD.

Diseño de Interfaz de Usuario.

The image shows a web form titled "Registrar Organismo" with a blue header. The form is organized into several sections:

- Órgano de Seguridad:** A dropdown menu.
- Organismo:** A section containing:
 - Nombre:** A text input field.
 - Dirección:** A sub-section with a grid of fields:
 - Estado:** Dropdown menu.
 - Municipio:** Dropdown menu.
 - Parroquia:** Dropdown menu.
 - Avenida:** Text input field.
 - Calle:** Text input field.
 - Caserío:** Text input field.
 - Aldea:** Text input field.
 - Urbanización:** Text input field.
 - Piso:** Text input field.
- Teléfono:** A text input field.
- Fotos:** A section with a text input field, a "... Otra" link, and a list of four checkboxes.
- Recursos a Gestionar:** A section with two large empty rectangular boxes and two navigation buttons: ">" and "<".

At the bottom right of the form are two buttons: "Aceptar" and "Cancelar".

Figura 18. Registrar Organismo.

Precondiciones.

No aplica.

Tabla 10: Flujo básico del CU Registrar Organismo.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona la opción “Registrar Organismo”.	2. Obtiene de la BD la lista de los órganos de seguridad y los tipos de recurso que existen.
	3. Muestra el formulario Registrar Organismo.
4. El Administrador de Recursos introduce los datos: <ul style="list-style-type: none"> • Órgano de seguridad. • Nombre. • Dirección. • Teléfono. • Fotos. • Recursos a gestionar: <ul style="list-style-type: none"> • Tipo de Recurso. 	
5. El Administrador de Recursos selecciona la opción “Aceptar”.	6. Comprueba que los datos introducidos son correctos.
	7. Comprueba que ese organismo no esté registrado en la BD.
	8. Almacena la información introducida en la BD.
Flujo Alterno 3ª Opción “Cancelar”	
Acción del Actor	Respuesta del Sistema
3ª.a.1. El Administrador de Recursos selecciona la opción “Cancelar”.	3ª.a.2. Retorna a la página anterior.
Flujo Alterno 6a Datos Introducidos Incorrectos	
Acción del Actor	Respuesta del Sistema
	6a.1. Comprueba que los datos introducidos

	son incorrectos.
	6a.2. Indica los datos que están incorrectos.
Flujo Alterno 7a Organismo Registrado Anteriormente	
Acción del Actor	Respuesta del Sistema
	7a.1. Comprueba que ese organismo ya está registrado en la BD.
	7a.2. Muestra un mensaje indicando que el organismo se encuentra registrado.
Referencias.	R16

CU Mostrar Organismo.

Propósito.

Mostrar el listado de organismos que pertenecen a un Órgano de Seguridad determinado.

Descripción.

Se obtiene de la BD y muestran los organismos que cumplan con los criterios de búsqueda definidos.

Diseño de Interfaz de Usuario.



Figura 19. Buscar Organismo.

Precondiciones.

No aplica.

Tabla 11: Flujo básico del CU Mostrar Organismo.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona la opción “Mostrar Organismo”.	2. Obtiene de la BD el listado de los órganos de seguridad. Determina un elemento de esa lista para mostrarlo por defecto.
	3. Obtiene de la BD el listado de los organismos usando el código del órgano de seguridad que se definió por defecto.
	4. Muestra el formulario Buscar Organismo.
5. El Administrador de Recursos selecciona el órgano de seguridad.	6. Obtiene de la BD, usando el código del órgano de seguridad seleccionado, los organismos que pertenecen al mismo.
	7. Muestra los organismos obtenidos.
8. El Administrador de Recursos selecciona el organismo.	
9. El Administrador de Recursos selecciona la opción “Buscar”.	10. Obtiene de la BD los datos del organismo usando los criterios de búsqueda introducidos. Estos datos son: <ul style="list-style-type: none"> • Código del organismo. • Nombre del organismo. • Teléfono.
	11. Comprueba que la búsqueda obtuvo resultados.
	12. Muestra los siguientes datos de los organismos obtenidos: <ul style="list-style-type: none"> • Nombre. • Teléfono.
	13. Habilita las opciones:

	<ul style="list-style-type: none"> • “Modificar”. • “Eliminar”. • “Ver Detalles”.
Flujo Alternativo 11a No Existen Organismos	
Acción del Actor	Respuesta del Sistema
	11a.1. Comprueba que la búsqueda no obtuvo resultados.
	11a.2. Muestra un mensaje indicando que no existen organismos con esos criterios de búsqueda.
Referencias.	R17

CU Ver Detalles de Organismo.

Propósito.

Mostrar los detalles de un organismo.

Descripción.

Obtiene de la BD los datos del organismo seleccionado usando el código del mismo. Muestra los datos obtenidos.

Diseño de Interfaz de Usuario.



Figura 20. Ver Detalles de Organismo.

Precondiciones.

Debe estar mostrada la lista de organismos.

Tabla 12: Flujo básico del CU Ver Detalles de Organismo.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona un organismo y la opción para ver sus detalles.	2. Obtiene de la BD, usando el código del organismo seleccionado, los siguientes datos: <ul style="list-style-type: none"> • Nombre del organismo. • Órgano de seguridad. • Dirección. • Teléfono. • Lista de fotos. • Lista de recursos que gestiona.
	3. Muestra los datos obtenidos.

Referencias.	R18
---------------------	-----

CU Modificar Organismo.

Propósito.

Modificar los datos de un organismo.

Descripción.

El Administrador de Recursos selecciona el elemento que desea modificar. Usando el código del organismo seleccionado se muestra un formulario con los datos asociados al organismo de forma editable. El Administrador de Recursos modifica los datos. Se actualizan los datos en la BD.

Diseño de Interfaz de Usuario.

Modificar Organismo

Órgano de Seguridad
[Dropdown]

Organismo
Nombre
[Text Input]

Dirección
Estado [Dropdown] Municipio [Dropdown] Parroquia [Dropdown]
Avenida [Text Input] Calle [Text Input] Caserío [Text Input]
Aldea [Text Input] Urbanización [Text Input] Piso [Text Input]

Teléfono [Text Input]

Fotos [File Upload] [Otra](#)
[Image Placeholder] [Image Placeholder] [Image Placeholder] [Image Placeholder]

Recursos a Gestionar
[Empty Box] [Empty Box] [Navigation Arrows]

Aceptar Cancelar

Figura 21. Modificar Organismo.

Precondiciones.

Debe estar mostrada la lista de organismos.

Tabla 13: Flujo básico del CU Modificar Organismo.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona un organismo de la lista mostrada.	
2. El Administrador de Recursos selecciona la opción “Modificar”.	3. Obtiene de la BD, usando el código del organismo seleccionado, los siguientes datos: <ul style="list-style-type: none"> • Nombre del organismo. • Órgano de seguridad. • Dirección. • Teléfono. • Lista de fotos. • Lista de recursos que gestiona.
	4. Obtiene la lista de órganos de seguridad y la lista de tipos de recursos existentes asociados al organismo.
	5. Muestra el formulario Modificar Organismo.
6. El Administrador de Recursos modifica los datos necesarios.	
7. El Administrador de Recursos selecciona la opción “Aceptar”.	8. Comprueba que los datos introducidos son correctos.
	9. Actualiza los datos en la BD usando el código del organismo seleccionado.
Flujo Alternativo 5ª Opción “Cancelar”	
Acción del Actor	Respuesta del Sistema
5ª.a.1. El Administrador de Recursos selecciona la opción “Cancelar”.	5ª.a.2. Retorna a la página anterior.

Flujo Alternativo 8a Datos Introducidos Incorrectos	
Acción del Actor	Respuesta del Sistema
	8a.1. Comprueba que los datos introducidos son incorrectos.
	8a.2. Indica los datos incorrectos.
Referencias.	R19

CU Eliminar Organismo.

Propósito.

Eliminar un organismo del sistema.

Descripción.

El Administrador de Recursos selecciona uno o varios organismos para eliminar. Se le muestra un mensaje de confirmación de eliminación. El Administrador de Recursos acepta la eliminación. Se eliminan de la BD los organismos seleccionados.

Diseño de Interfaz de Usuario.



Figura 22. Buscar Organismo.

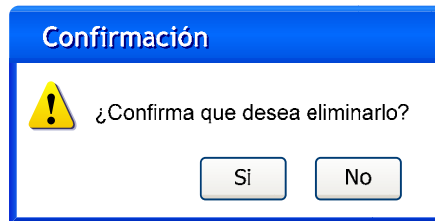


Figura 23. Confirmación de eliminación de Organismo.

Precondiciones.

Debe estar mostrada la lista de organismos.

Tabla 14: Flujo básico del CU Eliminar Organismo.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona uno o varios organismos.	
2. El Administrador de Recursos selecciona la opción "Eliminar".	3. Muestra un mensaje de confirmación de eliminación.
4. El Administrador de Recursos selecciona la opción "Si".	5. Marca como eliminado los datos asociados al (o los) organismo(s) seleccionado(s) usando el código de los mismos.
	6. Actualiza la lista de organismos mostrada.
Flujo Alternativo 4a Opción "No"	
Acción del Actor	Respuesta del Sistema
4a.1. El Administrador de Recursos selecciona la opción "No".	4a.2. Cierra el mensaje mostrado.
Referencias.	R20

CU Registrar Recurso.

Propósito.

Registrar un recurso en el sistema.

Descripción.

A partir de la categoría y del tipo de recurso seleccionado se crea un formulario con los datos que se deben insertar. Posteriormente se permite seleccionar el órgano de seguridad deseado y dentro de éste al organismo. Para seleccionar el organismo, si el usuario autenticado es del Centro 171, se le mostrará el listado de todos los organismos para que él seleccione a qué organismo pertenece el recurso. Si el usuario es de un organismo externo, en la lista de organismos sólo se mostrará el organismo del usuario autenticado.

Tabla 15: Flujo básico del CU Registrar Recurso.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo 171 selecciona la opción "Registrar Recurso".	2. Obtiene de la BD las categorías y tipos de recursos existentes.
	3. Muestra el formulario Registrar Recurso con los siguientes datos: <ul style="list-style-type: none"> • Categoría. • Tipo de recurso.
4. El Especialista de Recursos del Organismo 171 selecciona una categoría.	5. Obtiene de la BD, usando el código de la categoría seleccionada, los tipos de recursos asociados a esa categoría.
6. El Especialista de Recursos del Organismo 171 selecciona un tipo de recurso a insertar.	7. Obtiene de la BD los atributos del tipo de recurso seleccionado y el tipo de dato de cada atributo.
	8. Comprueba que el usuario autenticado pertenece al organismo 171.
	9. Obtiene de la BD y muestra los órganos registrados en el sistema.
10. El Especialista de Recursos del Organismo 171 selecciona el órgano.	11. Obtiene de la BD el listado de organismos pertenecientes al órgano seleccionado.
12. El Especialista de Recursos del Organismo 171 selecciona el organismo.	13. Construye un formulario usando los atributos obtenidos y el tipo de dato de cada atributo
	14. Muestra el formulario creado.
15. El usuario introduce los datos solicitados.	
16. El Especialista de Recursos del	17. Comprueba que los datos introducidos son

Organismo 171 selecciona la opción “Aceptar”.	correctos.
	18. Comprueba en la BD que el recurso no esté registrado usando los datos de los atributos que identifican unívocamente al tipo de recurso. Por ejemplo: la placa si es un vehiculo, el nombre y dirección si es un edificio, el numero de serie si un accesorio, etc.
	19. Almacena los datos introducidos en la BD. Si es un recurso de tipo Móvil se almacena por defecto el estado como habilitado.
Flujo Alterno 3*a Opción “Cancelar”	
Acción del Actor	Respuesta del Sistema
3*a.1. El Especialista de Recursos del Organismo 171 selecciona la opción “Cancelar”.	3*a.2. Retorna a la página anterior.
Flujo Alterno 8a El usuario autenticado es de un organismo externo	
Acción del Actor	Respuesta del Sistema
	8a.1. Comprueba que el usuario autenticado es de un organismo externo. 8a.2. Muestra el Órgano y el organismo al que pertenece el usuario. Ir a la acción 13.
Flujo Alterno 17a Datos Introducidos Incorrectos	
Acción del Actor	Respuesta del Sistema
	17a.1. Comprueba que los datos introducidos son incorrectos.
	17a.2. Indica los datos que están incorrectos.
Flujo Alterno 18a Recurso Registrado Anteriormente	

Acción del Actor	Respuesta del Sistema
	18a.1. Comprueba en la BD que el recurso está registrado usando los datos de los atributos que identifican unívocamente al tipo de recurso.
	18a.2. Muestra un mensaje indicando que el recurso ya se encuentra registrado.
Referencias.	R21

CU Mostrar Recurso.

Propósito.

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

Descripción.

Se obtiene de la BD los recursos que cumplan con los criterios de búsqueda definidos. Muestra el listado de recursos obtenido.

Diseño de Interfaz de Usuario.

The screenshot shows a web application window titled "Mostrar Recursos". It features a search section with the following elements:

- Criterios de Búsqueda:**
 - Categoría:** A dropdown menu.
 - Tipo de Recurso:** A dropdown menu.
 - Color:** A text input field.
 - Marca:** A dropdown menu.
 - Modelo:** A dropdown menu.
 - Matricula:** A text input field.
- Buscar:** A button to execute the search.
- Resultados de la Búsqueda:** A table with the following data:

	Categoría	Recurso	Matricula
<input checked="" type="radio"/>	Móvil	Ambulancia	VHG 346
<input type="radio"/>			
<input type="radio"/>			
- Acciones:** A row of links: [Modificar](#), [Eliminar](#), [Asociar Accesorios](#), [Mantenimiento](#), and [Expediente](#).

Figura 25. Buscar Recurso.

Precondiciones.

No aplica.

Tabla 16: Flujo básico del CU Mostrar Recurso.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo 171 selecciona la opción "Mostrar Recurso".	2. Obtiene de la BD el listado de las categorías y muestra la primera como valor por defecto.
	3. Obtiene de la BD el listado de tipos de

	recursos usando el código de la categoría que se definió por defecto.
	<p>4. Muestra la interfaz Buscar Recursos solicitando los siguientes criterios de búsqueda:</p> <ul style="list-style-type: none"> • Categoría. • Tipo de Recurso. <p>El campo categoría se va a mostrar con el valor definido por defecto y el tipo de recurso tomará los valores del listado obtenido.</p>
5. El Especialista de Recursos del Organismo 171 selecciona la categoría de recurso.	6. Obtiene de la BD y muestra, usando el código de la categoría seleccionada, los tipos de recursos.
	7. Muestra los tipos de recursos en el campo tipo.
8. El Especialista de Recursos del Organismo 171 selecciona el tipo de recurso.	9. Obtiene de la BD los atributos del tipo de recurso seleccionado.
	10. Muestra un campo para cada atributo obtenido dependiendo del tipo de dato del atributo.
11. El Especialista de Recursos del Organismo 171 introduce los criterios por los cuales va a buscar.	
12. El Especialista de Recursos del Organismo 171 selecciona la opción "Buscar".	<p>13. Obtiene de la BD usando los criterios de búsqueda introducidos el listado de recursos con el tipo de recurso y con los datos que lo identifican unívocamente en dependencia de la categoría. Estos datos son:</p> <ul style="list-style-type: none"> • Si la categoría por la que se buscó es Fijo

	<p>se obtiene:</p> <ul style="list-style-type: none"> • Dirección. <p>• Si la categoría por la que se buscó es Móvil se obtiene:</p> <ul style="list-style-type: none"> • Matrícula. <p>• Si la categoría por la que se buscó es Accesorio se obtiene:</p> <ul style="list-style-type: none"> • Número de serie.
	14. Comprueba que la búsqueda obtuvo resultado.
	15. Muestra el listado de los recursos obtenidos.
	<p>16. Habilita las opciones:</p> <ul style="list-style-type: none"> • “Ver Detalles de un Recurso”. • “Modificar”. • “Eliminar”. • “Mantenimiento”. • “Asociar Accesorios”. • “Expediente”. <p>Si la categoría de los recursos mostrados es “Accesorio” no se habilita la opción para asociar accesorios.</p>
Flujo Alternativo 14a No Existen Recursos	
Acción del Actor	Respuesta del Sistema
	14a.1. Comprueba que la búsqueda no obtuvo resultado.
	14a.2. Muestra un mensaje indicando que no existen recursos con esos criterios de búsqueda.
Referencias.	R22

CU Ver Detalles de Recurso.

Propósito.

Mostrar los detalles un recurso.

Descripción.

Se obtiene de la BD los datos del recurso seleccionado usando el código del mismo. Muestra los datos obtenidos.

Diseño de Interfaz de Usuario.



Figura 26. Detalles de Recurso (Ejemplo: Edificio).

Precondiciones.

Debe estar mostrada la lista de recursos.

Tabla 17: Flujo básico del CU Ver Detalles de Recurso.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del	2. Obtiene de la BD los atributos del tipo de

Organismo 171 selecciona la opción “Ver los detalles de un recurso”.	recurso seleccionado y el tipo de dato de cada atributo.
	3. Crea un formulario usando los atributos obtenidos y el tipo de dato de cada atributo.
	4. Obtiene de la BD los datos del recurso seleccionado usando el código del mismo.
	5. Muestra el formulario creado con los datos obtenidos.
Referencias.	R23

CU Modificar Recurso.

Propósito.

Modificar los datos relacionados a un recurso.

Descripción.

El Especialista de Recursos del Organismo 171 selecciona el elemento que desea modificar. Se crea el formulario según el tipo de recurso seleccionado con todos los datos asociados a este recurso de forma editable. El Especialista de Recursos del Organismo 171 modifica los datos. Se actualizan los datos en la BD.

Diseño de Interfaz de Usuario.

Figura 27. Modificar Recurso (Ejemplo: Edificio).

Precondiciones.

Debe estar mostrada la lista de recursos.

Tabla 18: Flujo básico del CU Modificar Recurso.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo 171 selecciona un recurso de la lista de recurso mostrada.	
2. El Especialista de Recursos del Organismo 171 selecciona la opción "Modificar".	3. Obtiene de la BD, usando el código del recurso seleccionado, los siguientes datos: <ul style="list-style-type: none"> • Atributos del tipo de recurso. • Tipo de dato de cada atributo.
	4. Crea un formulario usando el resultado de la búsqueda.
	5. Obtiene de la BD los datos del recurso seleccionado usando el código del mismo.
	6. Muestra el formulario creado con los datos del recurso y los campos editables.
7. El Especialista de Recursos del Organismo 171 modifica los datos necesarios.	
8. El Especialista de Recursos del Organismo 171 selecciona la opción "Aceptar".	9. Comprueba que los datos introducidos son correctos.
	10. Actualiza en la BD, usando el código del recurso, los datos modificados.
Flujo Alterno 6*a Cancelar	
Acción del Actor	Respuesta del Sistema
6*a.1. El Especialista de Recursos del Organismo 171 selecciona la opción "Cancelar".	6*a.2. Retorna a la página anterior.

Flujo Alterno 9a Datos Introducidos Incorrectos	
Acción del Actor	Respuesta del Sistema
	9a.1. Comprueba que los datos introducidos son incorrectos.
	9a.2. Indica los datos que están incorrectos. Los datos se mantienen para que el Especialista de Recursos del Organismo 171 los modifique si lo desea.
Referencias.	R24

CU Eliminar Recurso.

Propósito.

Eliminar un recurso del sistema.

Descripción.

El Especialista de Recursos del Organismo 171 selecciona uno o varios recursos para eliminar. Se le muestra un mensaje de confirmación de eliminación. El Especialista de Recursos del Organismo 171 acepta la eliminación y se marcan en la BD los recursos como eliminados.

Diseño de Interfaz de Usuario.

Mostrar Recursos

Criterios de Búsqueda

Categoría Tipo de Recurso

Color

Marca

Modelo

Matricula

Buscar

Resultados de la Búsqueda

	Categoría	Recurso	Matricula
<input checked="" type="radio"/>	Móvil	Ambulancia	VHG 346
<input type="radio"/>			
<input type="radio"/>			

[Modificar](#) [Eliminar](#) [Asociar Accesorios](#) [Mantenimiento](#) [Expediente](#)

Figura 28. Buscar Recurso.

Confirmación

⚠ ¿Confirma que desea eliminarlo?

Si No

Figura 29. Confirmación de Eliminación de Recursos.

Precondiciones.

Debe estar mostrada la lista de recursos.

Tabla 19: Flujo básico del CU Eliminar Recurso.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo 171 selecciona uno o varios recursos de la lista de recursos.	
2. El Especialista de Recursos del Organismo 171 selecciona la opción "Eliminar".	3. Muestra un mensaje de confirmación de eliminación de los recursos.
4. El Especialista de Recursos del Organismo 171 selecciona la opción "Si".	5. Elimina el o los recursos del sistema marcándolo como eliminado en la BD, usando el o los códigos de los recursos seleccionados.
	6. Actualiza la lista de recursos mostrada.
Flujo Alternativo 4a Opción "No"	
Acción del Actor	Respuesta del Sistema
4a.1. El Especialista de Recursos del Organismo 171 selecciona la opción "No".	4a.2. Cierra el mensaje mostrado.
Referencias.	R25

CU Asociar Accesorios a Recursos.

Propósito.

Asociar medios de comunicación a los recursos.

Descripción.

El Especialista de Recursos del Organismo 171 selecciona la opción "Asociar Accesorios" y el sistema crea una interfaz cuyos atributos editables dependen del tipo de recurso del recurso seleccionado. El Especialista de Recursos del Organismo 171 selecciona los accesorios que desea asociar y se almacenan esos datos en la BD.

Diseño de Interfaz de Usuario.



Figura 30. Asociar Accesorios.

Precondiciones.

Debe estar mostrada la lista de recursos.

Debe estar mostrada una categoría de recursos diferente de “Accesorios”.

Tabla 20: Flujo básico del CU Asociar Accesorios a Recursos.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo 171 selecciona un recurso.	
2. El Especialista de Recursos del Organismo 171 selecciona la opción “Asociar Accesorio”.	3. Obtiene de la BD el tipo de recurso del recurso seleccionado y los atributos que identifican unívocamente a ese tipo de recurso, así como el valor de esos atributos. Si el tipo de recurso pertenece a la categoría “Móvil” el atributo es:

	<p>Matrícula</p> <p>Si el tipo de recurso pertenece a la categoría “Fijo” los atributos variarán en función del tipo de recurso del recurso seleccionado.</p>
	4. Obtiene de la BD el listado de accesorios.
	5. Crea un formulario usando los atributos obtenidos con sus valores y carga la lista de accesorios posibles en el campo accesorio.
	6. Muestra el formulario creado.
7. El Especialista de Recursos del Organismo 171 selecciona el tipo de accesorio que desea asociar.	8. Obtiene de la BD los números de serie que no se encuentren asociados a un recurso usando el código del tipo de accesorio seleccionado.
	9. Muestra los números de series obtenidos.
10. El Especialista de Recursos del Organismo 171 selecciona el número de serie.	
11. El Especialista de Recursos del Organismo 171 selecciona la opción “Adicionar”.	12. Comprueba que el accesorio no se encuentra en la lista de accesorios ya introducida.
	13. Adiciona a la lista el tipo de accesorio y el número de serie.
14. El Especialista de Recursos del Organismo 171 selecciona la opción “Aceptar”.	15. Almacena en la BD el listado de accesorios asociados para el recurso seleccionado.
Nota: Las acciones del 7 al 13 se pueden repetir tantas veces como sea necesario.	
Flujo Alterno 6*a Opción “Cancelar”	
Acción del Actor	Respuesta del Sistema
6*a.1. El Especialista de Recursos del	6*a.2. Retorna a la página anterior.

Organismo 171 selecciona la opción “Cancelar”.	
Flujo Alternativo 12a Accesorio Encontrado	
Acción del Actor	Respuesta del Sistema
	12a.1. Comprueba que el accesorio se encuentra en la lista.
	12a.2. Muestra un mensaje indicando que ese accesorio está en la lista de los que se desean asociar.
Flujo Alternativo 14*a Opción “Eliminar”	
Acción del Actor	Respuesta del Sistema
14*a.1. El Especialista de Recursos del Organismo 171 selecciona un accesorio de la lista.	
14*a.2. El Especialista de Recursos del Organismo 171 selecciona la opción “Eliminar”.	14*a.3. Elimina de la lista el accesorio seleccionado usando su número de serie.
Referencias.	R26

CU Modificar Estado de Recurso.

Propósito.

Modificar el estado de un recurso de tipo móvil.

Descripción.

El Habilitador selecciona la opción habilitar o deshabilitar y el sistema le muestra una interfaz con los recursos de tipo móvil que se encuentren en estado habilitado o deshabilitado en dependencia de la opción seleccionada por el Habilitador. El Habilitador modifica el estado de los recursos. Se actualizan los estados en la BD.

Diseño de Interfaz de Usuario.



Figura 31. Modificar Estado de Recurso (Habilitar).

Precondiciones.

No aplica.

Tabla 21: Flujo básico del CU Modificar Estado de Recurso.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Habilitador selecciona la opción "Modificar Estado de Recurso".	2. Muestra el formulario Modificar Estado de Recurso.
3. El Habilitador selecciona la opción: Si selecciona la opción "Habilitar". Ir a la sección "Opción Habilitar". Si selecciona la opción "Deshabilitar". Ir a la sección "Opción Deshabilitar".	
Flujo Alternativo 2*a Cancelar	
Acción del Actor	Respuesta del Sistema
2*a.1. El Habilitador selecciona la	2*a.1. Retorna a la página anterior.

opción "Cancelar".	
Sección "Opción Habilitar"	
	4. Obtiene de la BD el listado de los recursos que están deshabilitados.
	5. Muestra una lista con: <ul style="list-style-type: none"> • Tipo de recurso. • Matrícula. • Hora de Habilitación.
6. El Habilitador selecciona los recursos que desea habilitar.	
7. El Habilitador selecciona la opción "Aceptar".	8. Actualiza el estado de los recursos en la BD.
Sección "Opción Deshabilitar"	
	4. Obtiene de la BD el listado de los recursos que están habilitados.
	5. Muestra una lista con: <ul style="list-style-type: none"> • Tipo de recurso. • Matrícula. • Hora de deshabilitación.
6. El Habilitador selecciona los recursos que desea deshabilitar.	
7. El Habilitador selecciona la opción "Aceptar".	8. Actualiza el estado de los recursos en la BD.
Referencias.	R30

CU Solicitar Recursos.

Propósito.

Registrar una solicitud de recursos.

Descripción.

Se muestra un formulario con los datos que se deben insertar relacionados con la solicitud. Se introducen los datos y se registra la solicitud en la BD. Los tipos de recurso a solicitar son aquellos que puede registrar el organismo al que se le solicita recurso.

Diseño de Interfaz de Usuario.

The screenshot shows a software window titled "Solicitud de Recursos". It contains the following elements:

- Asunto:** A single-line text input field.
- Descripción:** A multi-line text area with scrollbars.
- Fecha en que se necesitan los recursos:** A date selection dropdown menu.
- Órgano de Seguridad:** A dropdown menu.
- Organismo:** A dropdown menu.
- Listado de Recursos:** A section containing:
 - A dropdown menu for "Tipo de Recurso".
 - A spinner control for "Cantidad".
 - An "Adicionar" button.
 - A table with two columns: "Tipo de Recurso" and "Cantidad". The table currently has two empty rows, each with a red 'X' icon in the right margin.
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

Figura 32. Solicitud de Recursos.

Precondiciones.

No aplica.

Tabla 22: Flujo básico del CU Solicitar Recursos.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Recursos selecciona la opción "Solicitar Recursos".	2. Obtiene de la BD los órganos de seguridad.
	3. Muestra el formulario Solicitud de Recursos.
4. El Administrador de Recursos introduce el asunto, la descripción y fecha en que se necesitan los recursos.	
5. El Administrador de Recursos selecciona el órgano de seguridad.	6. Obtiene de la BD, usando el código del órgano de seguridad, los organismos asociados al órgano de seguridad seleccionado.
7. El Administrador de Recursos selecciona el organismo al que desea solicitar recursos.	8. Obtiene de la BD, usando el código del organismo, los tipos de recursos que puede registrar el organismo seleccionado.
9. El Administrador de Recursos selecciona el tipo de recurso y especifica la cantidad a solicitar.	
10. El Administrador de Recursos selecciona la opción "Adicionar".	11. Adiciona el tipo de recurso y la cantidad a la lista de solicitudes de recursos. Si el usuario ya había seleccionado ese tipo, incorpora la cantidad actual a la cantidad anterior que se hubiera especificado.
12. El Administrador de Recursos selecciona la opción "Aceptar".	13. Comprueba que los datos introducidos son correctos.
	14. Almacena en la BD la información introducida además de la hora en que se registra la solicitud de recursos y el código del

	usuario autenticado.
Nota: Las acciones desde la 9 hasta la 11 se pueden realizar tantas veces como necesario hasta haber seleccionados todos los recursos que se van a solicitar.	
Flujo Alternativo 3ª Opción "Cancelar"	
Acción del Actor	Respuesta del Sistema
3ª.a.1. El Administrador de Recursos selecciona la opción "Cancelar".	3ª.a.2. Retorna a la página anterior.
Flujo Alternativo 11ª Opción "Eliminar"	
Acción del Actor	Respuesta del Sistema
11ª.a.1. El Administrador de Recursos selecciona un tipo de recurso.	
11ª.a.2. El Administrador de Recursos selecciona la opción "Eliminar".	11ª.a.3. Elimina de la lista temporal el recurso seleccionado usando el tipo de recurso seleccionado.
Flujo Alternativo 13a Datos Introducidos Incorrectos	
Acción del Actor	Respuesta del Sistema
	13a.1. Comprueba que los datos introducidos son incorrectos.
	13a.2. Indica los datos que están incorrectos.
Referencias.	R27

CU Mostrar Solicitudes de Recursos.

Propósito.

Mostrar las solicitudes de recursos que cumplan con los criterios de búsqueda introducidos por el usuario.

Descripción.

Se obtiene de la BD el listado de solicitudes de recursos que cumplan con los criterios especificados.

Muestra el listado de solicitudes de recursos obtenido.

Diseño de Interfaz de Usuario.

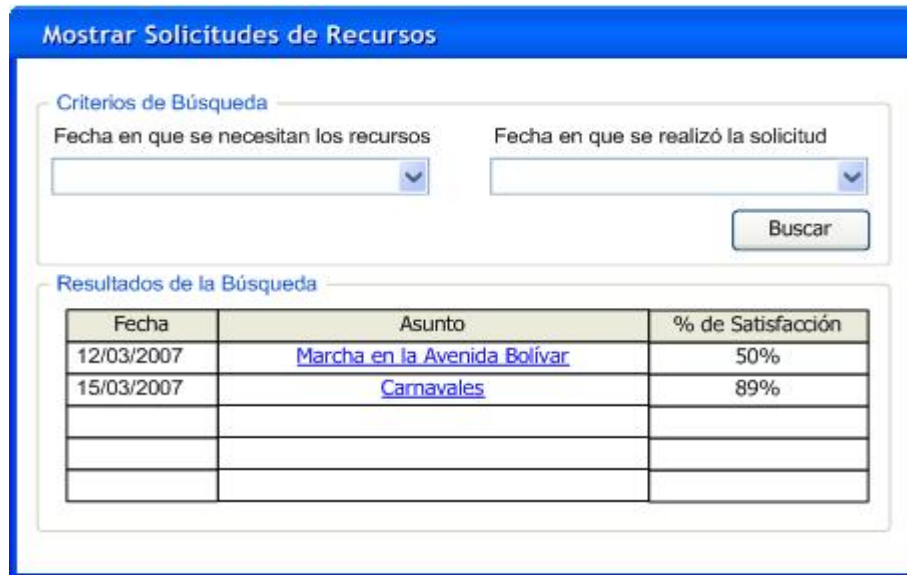


Figura 33. Buscar Solicitudes de Recursos.

Precondiciones.

No aplica.

Tabla 23: Flujo básico del CU Mostrar Solicitudes de Recursos.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo selecciona la opción "Mostrar solicitudes de recursos".	2. Muestra un formulario Buscar Solicitudes de Recursos.
3. El Especialista de Recursos del Organismo introduce los criterios de búsqueda: Fecha en que se necesitan los recursos. Fecha en que se realizó la solicitud.	
4. El Especialista de Recursos del	5. Comprueba que existen solicitudes de

Organismo selecciona la opción “Buscar”.	recursos que cumplan con los criterios de búsqueda para el organismo del usuario autenticado.
	<p>6. Obtiene de la BD, usando el código del organismo, los siguientes datos de las solicitudes que van dirigida a ese organismo:</p> <ul style="list-style-type: none"> • Asunto. • Fecha en que se necesitan los recursos. • Lista de recursos solicitados: <ul style="list-style-type: none"> ○ Tipo de recurso. ○ Cantidad solicitada. ○ Cantidad de recursos con el estado Habilitado que tiene el organismo.
	<p>7. Calcula el por ciento de satisfacción para cada una de las solicitudes.</p> <p>El por ciento se obtiene usando la cantidad total de recursos solicitados y la cantidad total de recursos otorgados y se calcula que por ciento representa la asignación con respecto a lo pedido.</p>
	<p>8. Muestra los siguientes datos de las solicitudes de recursos:</p> <ul style="list-style-type: none"> • Fecha en que se necesitan los recursos. • Asunto de la solicitud. • Por ciento de satisfacción.
	9. Habilita la opción para “Ver Detalles de una Solicitud de Recursos”.
Flujo Alterno 5a No Existen Solicitudes de Recursos	
Acción del Actor	Respuesta del Sistema

	5a.1. Comprueba que no existen solicitudes de recursos para el organismo del usuario autenticado.
	5a.2. Muestra un mensaje indicando que no existen solicitudes de recursos con esos criterios de búsqueda.
Referencias.	R28

CU Ver Detalles de Solicitud de Recursos.

Propósito.

Mostrar los detalles de una solicitud de recursos.

Descripción.

El Especialista de Recursos del Organismo selecciona una solicitud de recursos y se obtiene de la BD los datos de la solicitud de recurso seleccionada usando el código de la misma. Muestra los datos obtenidos.

Diseño de Interfaz de Usuario.

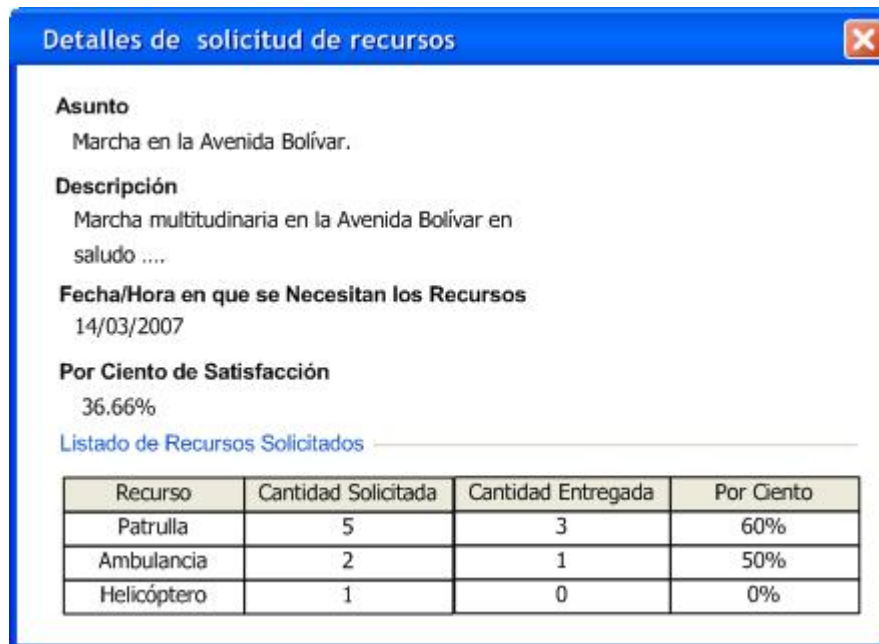


Figura 34. Detalles de Solicitud de Recursos.

Precondiciones.

Debe estar mostrada la lista de solicitudes de recursos.

Debe existir una solicitud de recursos seleccionada.

Tabla 24: Flujo básico del CU Ver Detalles de Solicitud de Recursos.

Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El Especialista de Recursos del Organismo selecciona la opción “Ver Detalles de una Solicitud de Recursos”.	2. Obtiene de la BD, usando el código de la solicitud de recursos seleccionada, los datos: <ul style="list-style-type: none"> • Asunto. • Descripción. • Fecha en que se necesitan los recursos. • Lista de recursos solicitados: <ul style="list-style-type: none"> ○ Tipo de recurso. ○ Cantidad solicitada. ○ Cantidad de recursos con el estado Habilitado que tiene el organismo.
	3. Calcula el por ciento de recursos entregados por cada tipo de recurso de la lista de solicitudes de recursos.
	4. Calcula el por ciento de satisfacción para la solicitud. El por ciento se obtiene usando la cantidad total de recursos solicitados y la cantidad total de recursos otorgados a la solicitud en general y se calcula que por ciento representa la asignación con respecto a lo pedido.
	5. Muestra los siguientes datos:

	<ul style="list-style-type: none"> • Asunto. • Descripción. • Fecha en que se necesitan los recursos. • Por ciento de satisfacción de la solicitud. • Lista de recursos solicitados: <ul style="list-style-type: none"> ○ Tipo de recurso. ○ Cantidad solicitada. ○ Cantidad entregada. ○ Por ciento de satisfacción del tipo de recurso solicitado.
Referencias.	R29

2.5 Conclusiones

En el desarrollo de este capítulo se hizo una descripción de la propuesta de solución, lo que ha permitido una mejor comprensión del contexto a automatizar y de las características y restricciones que deben existir en el sistema para cumplir con los requerimientos del cliente. Se realizó el Modelo IDEF, se definieron los requisitos funcionales que debe cumplir el sistema, se identificaron actores y casos de uso del sistema. 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 caso de uso.

A partir de este punto se puede comenzar la construcción de la solución de software propuesta, que será presentada en el próximo capítulo.

Capítulo III. Construcción de la solución propuesta

3.1 Introducción

En este capítulo se realizará el modelo de diseño del módulo de Administración y Control de Recursos, en el que se incluyen las diferentes clases y sus relaciones que conforman el diagrama de clases del diseño, el cual se dividió en paquetes para facilitar su comprensión.

3.2 Diagramas de clases del diseño

El diagrama de clases del diseño para el subsistema de Administración y Control de Recursos se dividió en paquetes para facilitar su comprensión. A continuación se muestra el diagrama de paquetes y seguido los distintos diagramas correspondientes a cada paquete.

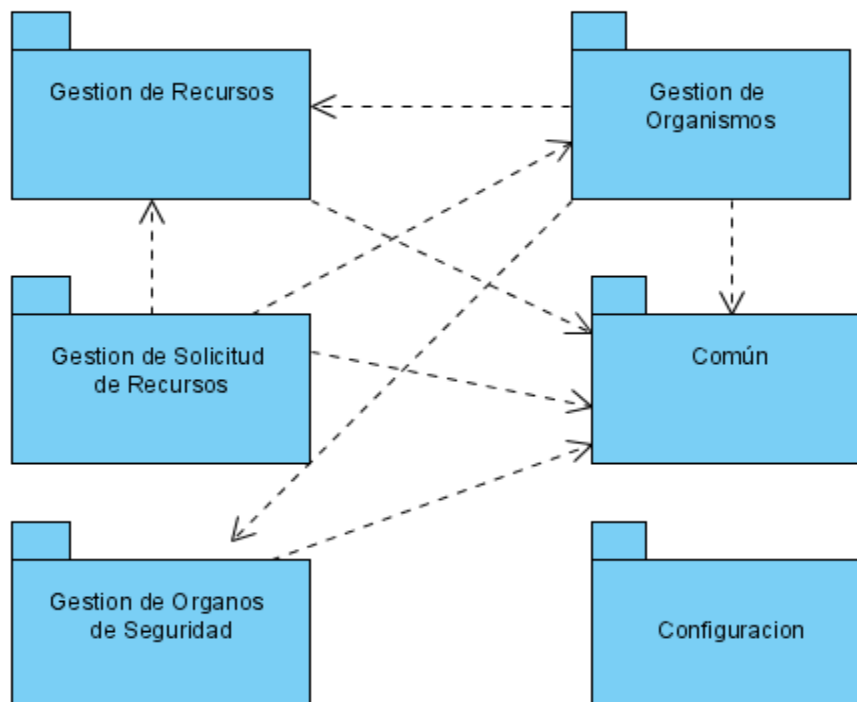


Figura 35: Diagrama de paquetes.

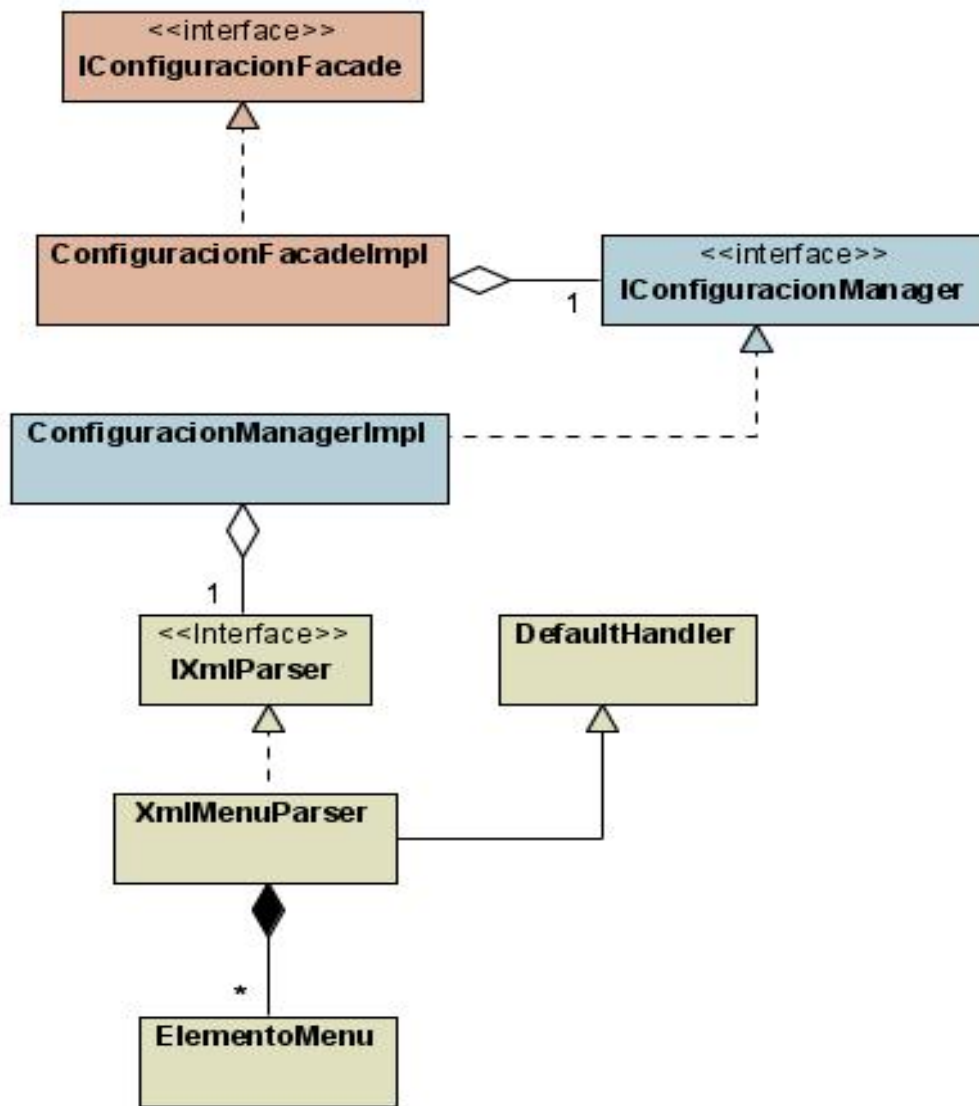


Figura 36: Diagrama de clases del diseño del Paquete Configuración.

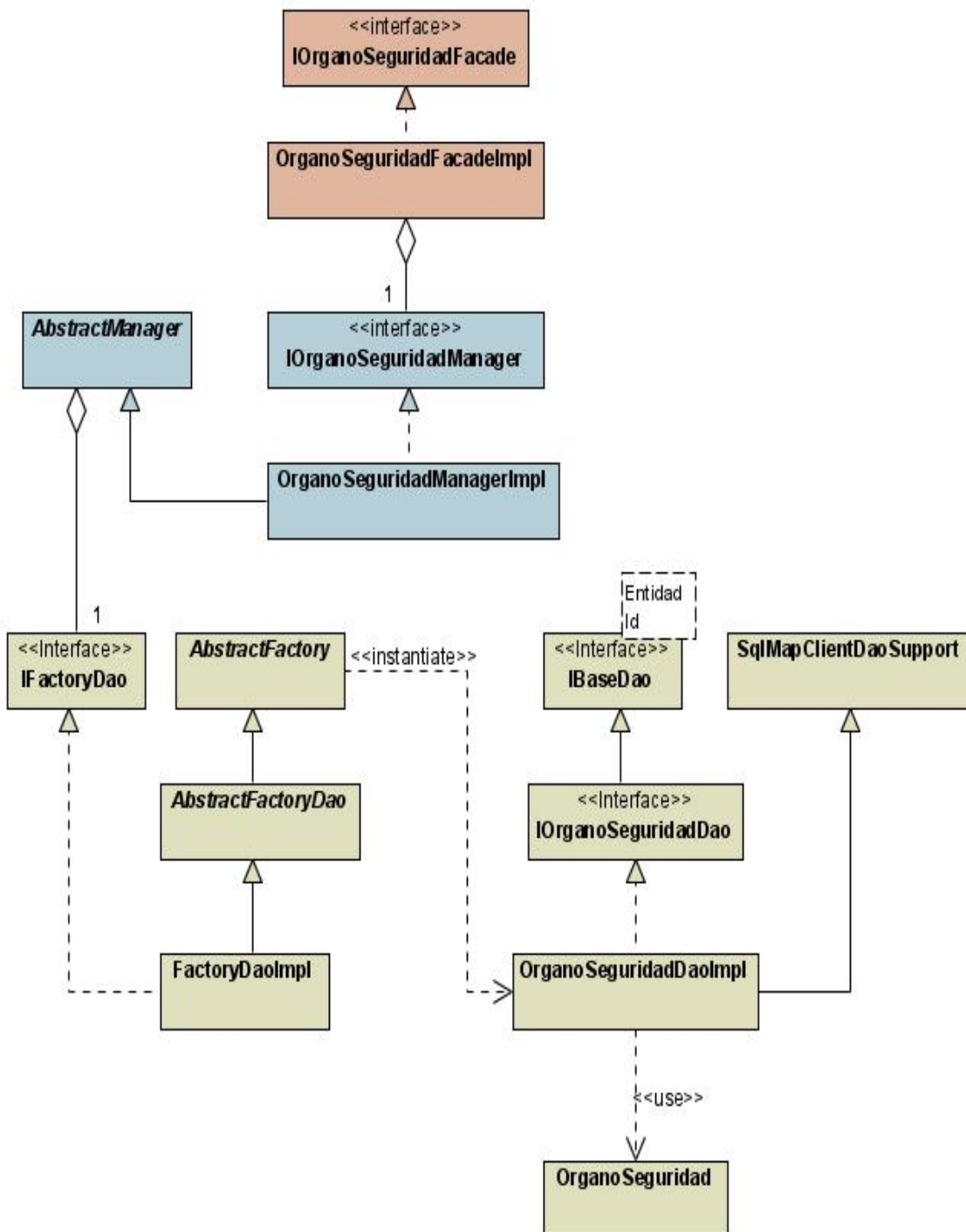


Figura 37: Diagrama de clases del diseño del Paquete Gestión de Órganos de Seguridad.

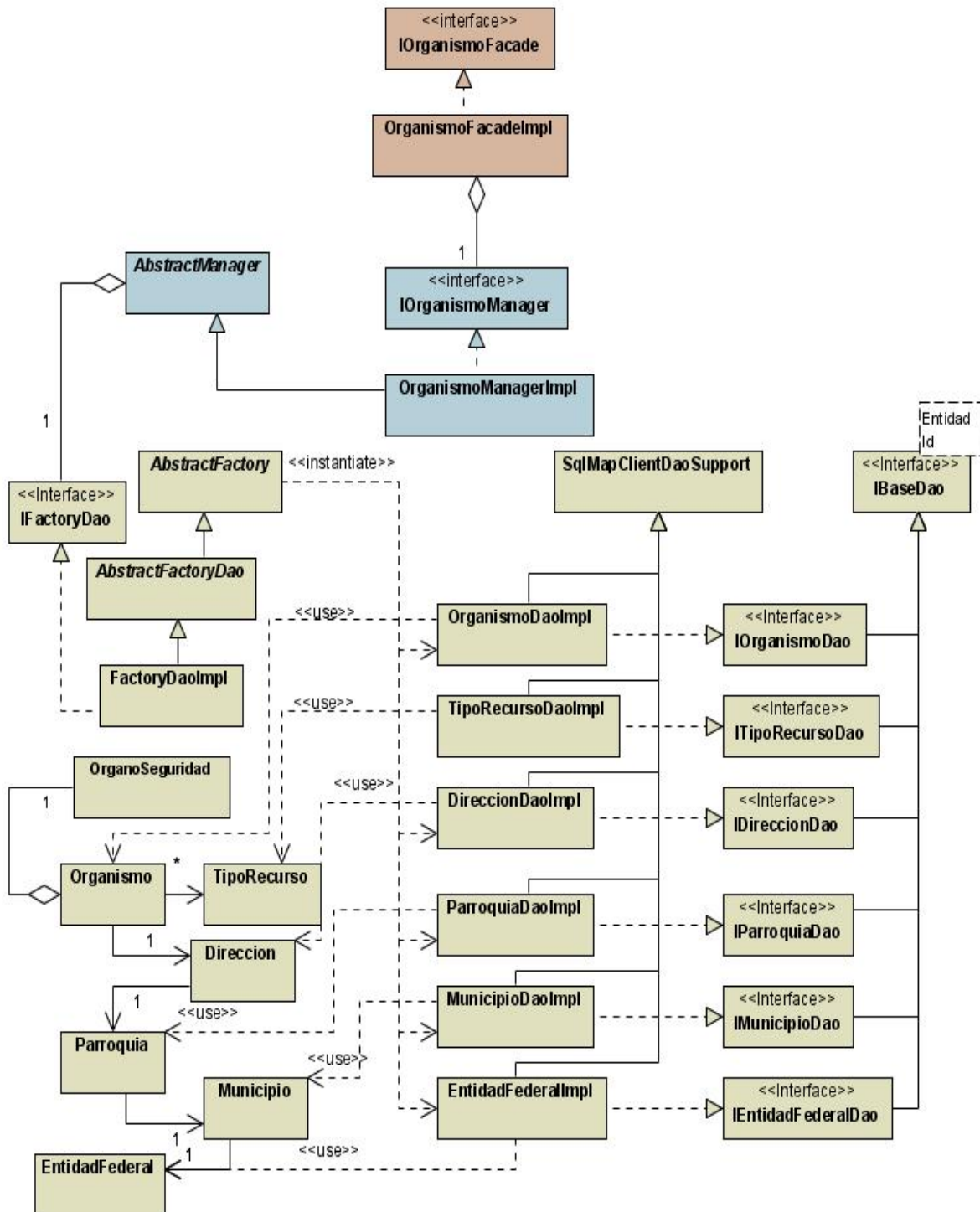


Figura 38: Diagrama de clases del diseño del Paquete Gestión de Organismos.

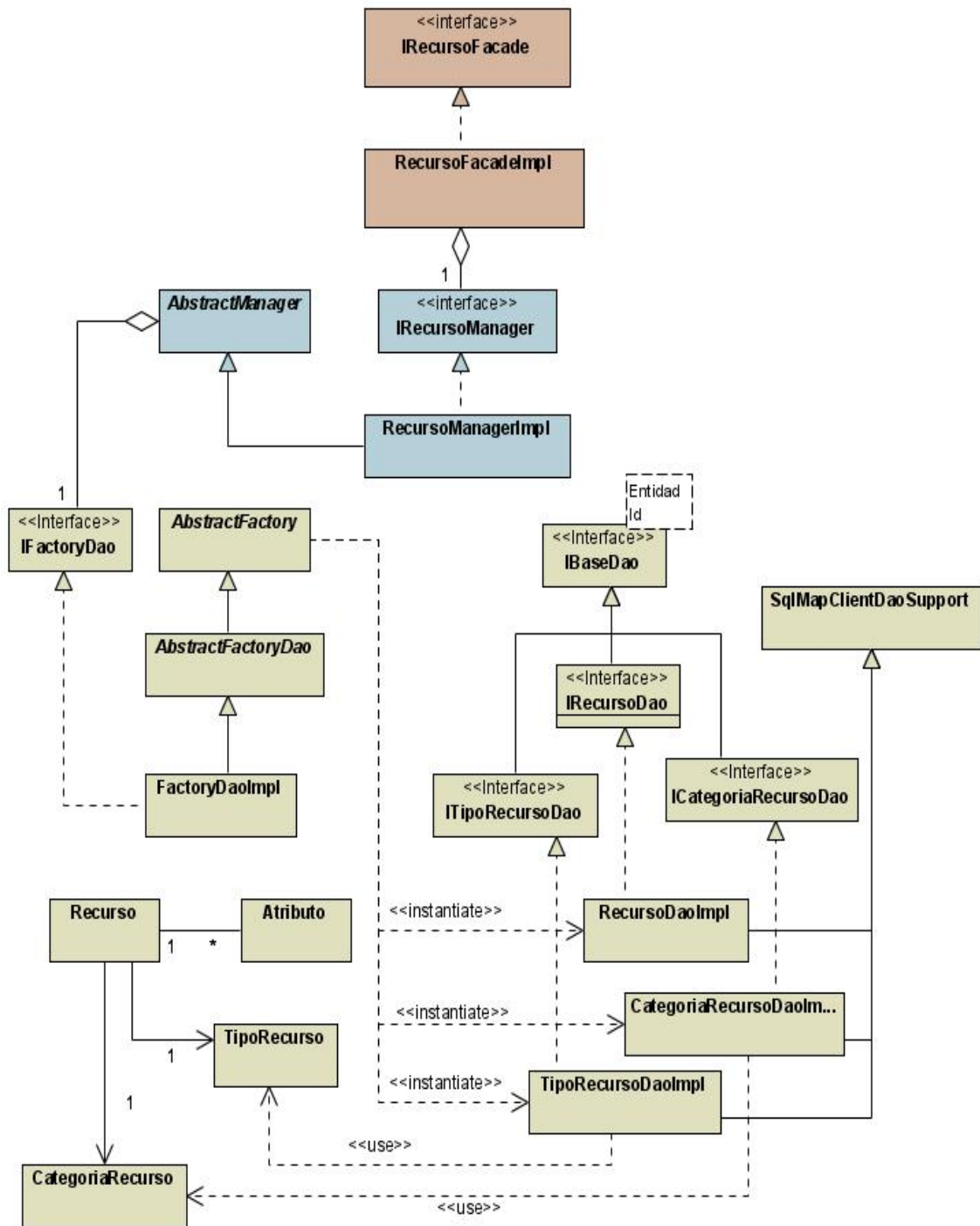


Figura 39: Diagrama de clases del diseño del Paquete Gestión de Recursos.

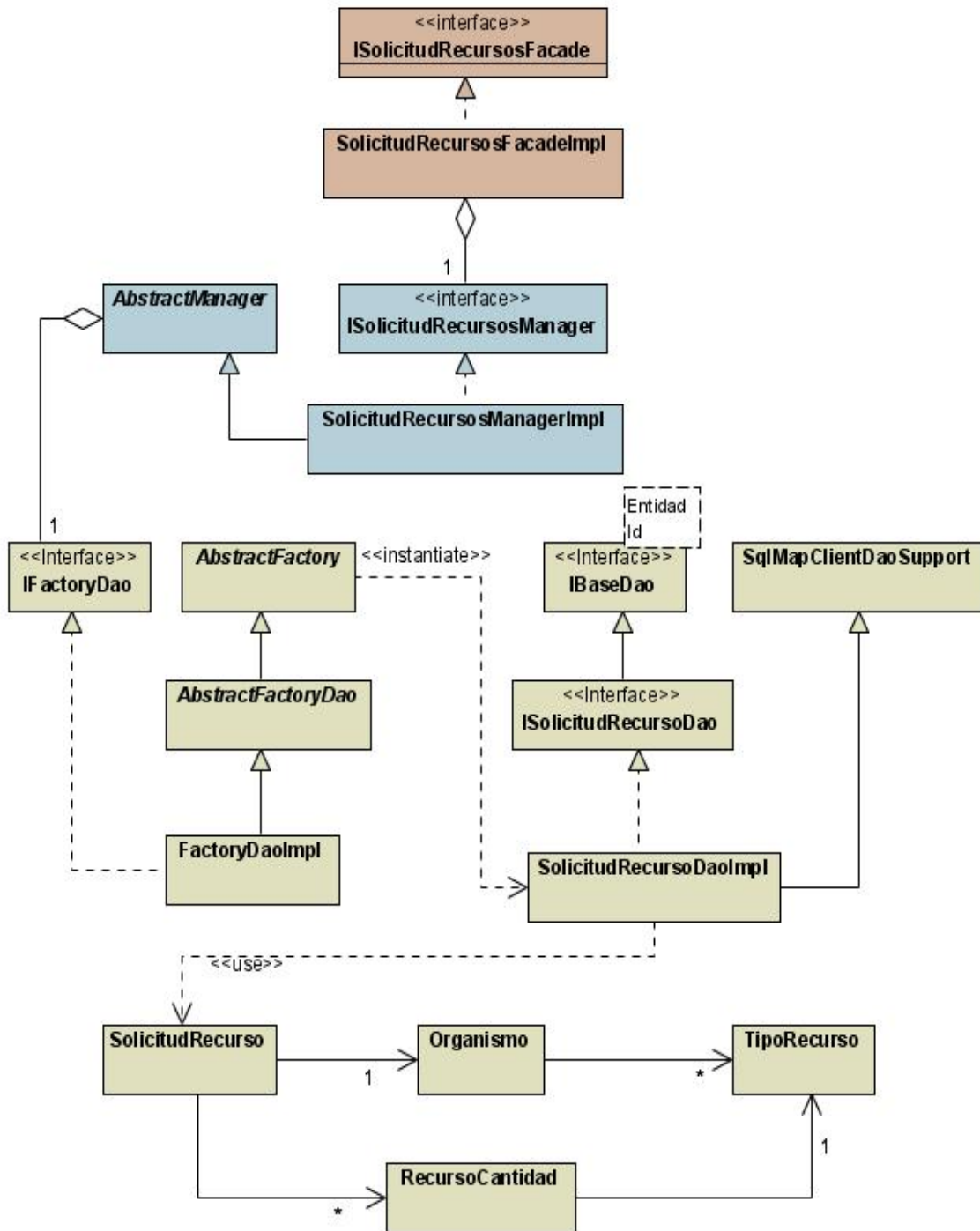


Figura 40: Diagrama de clases del diseño del Paquete Gestión de Solicitud de Recursos.

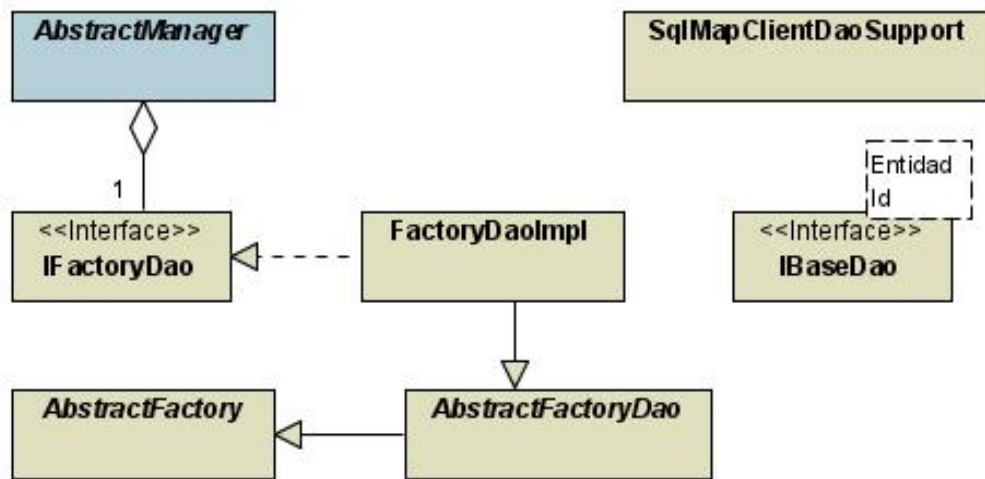


Figura 41: Diagrama de clases del diseño del Paquete Común.

3.3 Conclusiones

En este capítulo se modeló detalladamente la solución propuesta a través del modelo de diseño, para lo que se realizó el diagrama de clases, el cual se dividió en paquetes para facilitar la comprensión del mismo.

Conclusiones

Con el desarrollo de esta investigación se obtuvo el diseño de un sistema genérico y replicable que será utilizado para la administración y el control de los recursos en los Centros de Atención de Emergencias 171 en la República Bolivariana de Venezuela, cumpliéndose con el objetivo general definido. Además se puede concluir que:

- Se realizó un estudio de los sistemas informáticos existentes en Venezuela, así como las formas de administrar los recursos en los Centros de Atención de Emergencias 171 de este país.
- Se investigó sobre la situación de las tecnologías existentes con el objetivo de fundamentar las herramientas y métodos a utilizar.
- Se seleccionó la metodología RUP como la más adecuada para llevar a cabo todo el proceso de desarrollo.

Recomendaciones

Se recomienda implementar las funcionalidades del primer ciclo de desarrollo y continuar con el análisis, diseño e implementación, de las funcionalidades del segundo ciclo de desarrollo del Subsistema de Administración y Control de Recursos para el Centro de Gestión de Emergencias de Seguridad Ciudadana 171.

Referencias bibliográficas

1. GACETA OFICIAL DE LA REPUBLICA BOLIVARIANA DE VENEZUELA N° 37318 DEL 06 DE NOVIEMBRE DE 2001. [Consultado el 20/12/2006]; Disponible en <http://www.asambleanacional.gov.ve/ns2/leyes-habilitante/L-371SEGURIDADCIUDADANA.pdf>.
2. Ivar Jacobson, G.B., James Rumbaugh, *El proceso unificado de desarrollo de software*. 1998.
3. Patricio Letelier, M.C.P. (2006) *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)* **Volume**, DOI: 26.
4. Molpeceres, A. *Procesos de desarrollo: RUP, XP y FDD*. 2002 [Consultado el 8/01/2007]; Disponible en <http://www.javahispano.org/licencias/>.
5. *Microsoft Office System Product Information*. [Consultado el 8/06/ 2007]; Disponible en <http://www.microsoft.com/latam/office/visio/prodinfo/topten.mspx>.
6. Taylor, J. (2005) *Nuevas funciones y ventajas de Dreamweaver 8*. **Volume**, 5 [Consultado el 8/06/2007], Disponible en: http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures.html.
7. Jeff Ferguson, B.P., Jason Beres, Pierre Boutquin, Meeta Gupta., *La biblia de C#*. S.A. EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, Editor. 2003.
8. Subrahmanyam Allamaraju, C.B., John Davies, Tyler Jewell, Rod Johnson, Andy Longshaw, Ramesh Nagappan, Dr. P. G. Sarang, Alex Toussaint, Sameer Tyagi, Gary Watson, Marc Wilcox, Alan Williamson, *Programación Java Server con J2EE Edición 1.3*. 1206.

Bibliografía

IBM Rational Software [Consultado el: 8/06/2007]. Disponible en: <http://www-306.ibm.com/software/rational/>.

IDEF(Integrate Definition Methods) [Consultado el: 8/6/2007]. Disponible en: <http://www.idef.com/>.

Indudata, soluciones en Informática [Consultado el: 8/6/2007]. Disponible en: <http://www.indudata.com/mapa.htm>.

IVAR JACOBSON, G. B., JAMES RUMBAUG. El proceso unificado de desarrollo de software. 1998, Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>. ISBN 84-7829-036-2.

JAMES RUMBAUGH, I. J., GRADY BOOCH. El Lenguaje Unificado de Modelado. Manual de referencia. 1998, nº p. 528.

Java Tools: Visual Paradigm [Consultado el: 7/6/2007]. Disponible en: <http://javatoolbox.com/authors/visual-paradigm>.

JEFF FERGUSON, B. P., JASON BERES, PIERRE BOUTQUIN, MEETA GUPTA. La biblia de C#. Editado por: Ediciones Anaya Multimedia (Grupo Anaya, S. A. 2003, ISBN 84-415-1484-4.

Microsoft Office System Product Information [Consultado el: 8/06/2007]. Disponible en: <http://www.microsoft.com/latam/office/visio/prodinfo/topten.mspx>.

MOLPECERES, A. Procesos de desarrollo: RUP, XP y FDD Disponible en: <http://www.javahispano.org/licencias/>.

PATRICIO LETELIER, M. C. P. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP) 2006, vol. 05, Disponible en: <http://www.cyta.com.ar> , ISBN 1666-1680.

PRESSMAN, R. S. Ingeniería de Software. Un enfoque práctico. nº p. 614.

SUBRAHMANYAM ALLAMARAJU, C. B., JOHN DAVIES, TYLER JEWELL, ROD JOHNSON, ANDY LONGSHAW, RAMESH NAGAPPAN, DR. P. G. SARANG, ALEX TOUSSAINT, SAMEER TYAGI, GARY WATSON, MARC WILCOX, ALAN WILLIAMSON. Programación Java Server con J2EE Edición 1.3. 1206 p.

TAYLOR, J. Nuevas funciones y ventajas de Dreamweaver 8. 2005, 5 p. Disponible en:
http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures.html.

Visual Paradigm [Consultado el: 7/06/2007]. Disponible en: <http://www.visual-paradigm.com/>.

Web-Taller: Características de Dreamweaver [Consultado el: 8/6/2007]. Disponible en:
<http://www.web-taller.com.ar/dreamweaver.html>.

WENDY BOGGS, M. B. Mastering UML with Rational Rose 2002, ISBN 0-7821-4017-3.

Glosario de términos

Recurso: Todo aquel medio que puede ser utilizado para la atención de una emergencia, pueden ser móviles como ambulancias, carros de bomberos, patrullas, etc., accesorios que pueden ser radios, GPS, etc., y los fijos que pueden ser museos, escuelas, parques y otros.