

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL COMITÉ LOCAL PARA
LA PREVENCIÓN Y ATENCIÓN DE DESASTRES DE TULUÁ**



**DIANA LORENA VELANDIA VANEGAS
HERMAN ALFONSO VERA DELGADO**

**Trabajo de Grado para optar por el título de
Ingeniero de Sistemas**

**Director
Msc. JAIME LÓPEZ CARVAJAL**

**UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERIA DE SISTEMA Y COMPUTACIÓN
INGENIERÍA DE SISTEMAS
TULUÁ
2008**

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL COMITÉ LOCAL PARA
LA PREVENCIÓN Y ATENCIÓN DE DESASTRES DE TULUÁ**



**DIANA LORENA VELANDIA VANEGAS
HERMAN ALFONSO VERA DELGADO**

**Trabajo de Grado para optar por el título de
Ingeniero de Sistemas**

**Director
Msc. JAIME LÓPEZ CARVAJAL**

**UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERIA DE SISTEMA Y COMPUTACIÓN
INGENIERÍA DE SISTEMAS
TULUÁ
2008**

CONTENIDO

	Pág.
RESUMEN	7
INTRODUCCIÓN	8
1. PLANTEAMIENTO DEL PROBLEMA	9
2. JUSTIFICACIÓN	10
3. OBJETIVOS	11
3.1 OBJETIVO GENERAL	11
3.2 OBJETIVOS ESPECÍFICOS	11
4. MARCO DE REFERENCIA	12
4.1 MARCO TEÓRICO	12
4.1.1 Sistemas de Información	12
4.1.1.1 Actividades	12
4.1.1.2 Objetivos	13
4.1.1.3 Tipos	13
4.1.2 Aplicaciones Web	14
4.1.2.1 Patrón de Diseño	15
4.1.2.2 Java y J2SE	16
4.1.3 Aplicaciones Móviles	18
4.1.3.1 J2ME	18
4.1.3.2 WML	19
4.1.4 Sistema de Información Espacial	20
4.1.4.1 Funciones	20
4.1.4.2 Arquitectura de Aplicaciones Espaciales	21
4.1.4.3 Bases de datos espaciales	22
4.2 MARCO CONTEXTUAL	24
4.3 MARCO CONCEPTUAL	25
5. METODOLOGÍA	31
5.1 ETAPAS DE DESARROLLO	31
6. MODELO DE ANÁLISIS	33
6.1 DEFINICIÓN DE REQUERIMIENTOS	33
6.1.1 Requerimientos Funcionales	33
6.1.2 Requerimientos no Funcionales	43
6.2 DEFINICIÓN DE CASOS DE USO	45
6.2.1 Diagramas de Casos de Uso	45
6.3 DIAGRAMAS DE SECUENCIA	51
6.3.1 Aplicación Web General	51
6.3.2 Aplicación de respaldo (J2ME)	58
6.3.3 Aplicación Espacial	64
7. MODELO DE DISEÑO	68
7.1 MODELOS DE NAVEGACIÓN	68
7.2 DISEÑO DE BASE DE DATOS	76
7.2.1 Modelo Conceptual – Modelo E/R	76
7.3 DISEÑO DE CLASES	77
7.3.1 Diagrama de Clases	77

7.3.1.1 Aplicación General	77
7.3.1.2 Aplicación Móvil	78
7.3.1.3 Aplicación Espacial	78
7.4 DIAGRAMAS DE ACTIVIDADES	79
8. IMPLEMENTACIÓN	81
8.1 SIWEPAD GENERAL	82
8.2 SIWEPAD ESPACIAL	85
8.2.1 Visualización de Mapas	86
8.2.2 Manipulación del Mapa	86
8.3 SIWEPAD MÓVIL DE RESPALDO	87
9. PRUEBAS Y/O RESULTADOS	88
9.1 ÁMBITO	88
9.2 PROCEDIMIENTO Y RESULTADOS OBTENIDOS	89
RECOMENDACIONES	92
PROYECCIONES	93
CONCLUSIONES	94
BIBLIOGRAFÍA	95

LISTA DE FIGURAS

	Pág.
Figura 1. Flujo de actividades de un SI	13
Figura 2. Capas de una aplicación Web	14
Figura 3. Componentes MVC	15
Figura 4. Arquitectura J2ME	18
Figura 5. Gateway WAP	20
Figura 6. Modelo básico de funcionamiento de una aplicación cartográfica en la web.	21
Figura 7. Arquitectura de cliente delgado	21
Figura 8. Arquitectura de cliente grueso	22
Figura 9. Etapas de desarrollo del proceso de ingeniería Web	31
Figura 10. Sistema SIWEPAD	81
Figura 11. Elementos de modelo SIWEPAD general	83
Figura 12. Elementos de vista SIWEPAD general	84
Figura 13. Elementos de control SIWEPAD general	84
Figura 14. Elementos de almacenamiento SIWEPAD general	85
Figura 15. Arquitectura SIWEPAD espacial	85
Figura 16. Componentes principales, arquitectura SIWEPAD móvil	87

ANEXOS

- ANEXO A.** Recursos y Presupuesto
- ANEXO B.** Estructura de Desglose de Trabajo (EDT)
- ANEXO C.** Casos de Uso Extendidos
- ANEXO D.** Diagramas de Secuencia Adicionales
- ANEXO E:** Modelos de Navegación Adicionales
- ANEXO F:** Modelo Lógico base de datos. Modelo Relacional
- ANEXO G:** Descripción de las Clases
- ANEXO H:** Archivo Map
- ANEXO I:** Algunos Prototipos Finales de Interfaz de la Aplicación
- ANEXO J:** Prototipos Generales de la Aplicación

RESUMEN

Desarrollo de un sistema de información Web para el manejo de información durante el proceso de atención de desastres ocurridos en el municipio de Tuluá, procesos que son llevados a cabo por los integrantes del Comité Local para la Prevención y Atención de Desastre (CLOPAD) de la localidad.

El sistema está constituido por una aplicación Web que permite registrar y consultar información de los desastres, de las viviendas, familias y personas afectadas por los mismos, así como de recursos y elementos necesarios para la atención y/o reconstrucción de los entes afectados; una aplicación que permite el registro y consulta, de la información mencionada, a través de dispositivos móviles y una aplicación espacial que permite la consulta de eventos de desastres y la visualización de estos en un mapa.

INTRODUCCIÓN

Los desastres tienen la posibilidad de presentarse en cualquier parte. En el momento en que un evento así ocurre, la situación es crítica por la magnitud de los problemas y las demandas urgentes que se generan en la población, donde las acciones de primera instancia son las de atender a las personas afectadas y capturar la información general del evento ocurrido.

Una actividad que es de vital importancia, las primeras horas después de ocurrido el desastre, es entonces, el proceso de evaluación de las necesidades que se realiza en la zona del desastre con el fin de determinar el tipo y la extensión de los efectos de la población afectada, los daños y las áreas que requieren ser intervenidas con mayor urgencia.

Por esto, la información que se maneja en el nivel local es importante y debe ser válida, clara, oportuna y expresada de forma sencilla, ya que sustenta la toma de decisiones de todo el sistema de atención de desastres.

El trabajo que se presenta tuvo como propósito, desarrollar un sistema de información bajo la Web que permite la gestión de información en el Comité Local para la Prevención y Atención de Desastres (CLOPAD). Igualmente, pretende apoyar a la comunidad, y a los evaluadores del comité en particular, con una herramienta que pueda ser utilizada en situaciones de desastre en las labores de atención y/o evaluación de daños y análisis de necesidades (EDAN).

Por otro lado, se desarrolló una aplicación móvil que permite el registro de la información de mayor relevancia en el lugar del desastre, esto, con el fin de facilitar la captura de datos en sitio, dado que las condiciones de este no siempre son las más adecuadas.

Finalmente, se adecuó el sistema con el fin de permitir visualizar información de desastres a través de un mapa en el cual el CLOPAD, la comunidad en general y demás entidades interesadas, puedan consultar de manera gráfica los desastres que ocurren en la ciudad.

El sistema mencionado es de gran ayuda para los miembros del CLOPAD, ya que el acceso y almacenamiento de la información más relevante acerca de los desastres y los daños que este ocasiona, se puede realizar desde cualquier lugar, logrando así el mejoramiento en la calidad de la atención de personas afectadas.

Para la construcción de dicho proyecto se contó con tecnologías de desarrollo de aplicaciones y de procesamiento de información de dominio público como: lenguajes de programación, sistemas de bases de datos, servidores de aplicaciones Web y de mapas.

1. PLANTEAMIENTO DEL PROBLEMA

Actualmente el Comité Local para la Prevención y Atención de Desastres (CLOPAD) de la ciudad de Tuluá, realiza la toma de información en la evaluación de daños y análisis de necesidades (EDAN) de forma manual, lo que genera que esta sea difícil de procesar y analizar de manera rápida y que se dificulte el control de los datos ocasionando que se presente duplicidad de información, como el múltiple registro de una persona damnificada en un mismo desastre o en diferentes, ocurridos en una misma fecha. Además, se utilizan formularios impresos lo que ocasiona que personas ajenas al comité puedan tener acceso a ellos y realizar su diligenciamiento y que las condiciones climáticas y/o el tiempo deterioren el papel y por ende la información allí contenida. Esto hace que la información sea no confiable, incompleta y algunas veces confusa. Igualmente, al momento de establecer las estrategias de acción en la mitigación, no se dispone rápidamente con información de los recursos existentes (personas, equipos, insumos, mapas, etc.).

Lo anterior, dificulta la evaluación de daños y la determinación de necesidades frente a un desastre ocurrido, y hace complejo tener la información apropiada para establecer las decisiones adecuadas y oportunas que ayuden en la atención o mitigación de un determinado desastre. Además, hace que se presenten dificultades al momento de realizar la evaluación de forma cualitativa y cuantitativa de los daños que ha ocasionado un evento adverso.

La situación planteada genera consecuencias que afectan directamente a la población, sumándose a ello la escasez de apoyo económico lo que acentúa la vulnerabilidad de la comunidad causando pérdidas de vidas, de bienes materiales, mala calidad de vida, entre otros; efectos que se pudieran reducir a través de una plataforma unificada que incorpore toda la información relevante de un evento ocurrido.

2. JUSTIFICACIÓN

En el CLOPAD de Tuluá, el proceso de evaluación de daños y análisis de necesidades se ha tornado complejo y lento, dadas las características de la información que se maneja y que su registro y almacenamiento se hace de forma manual.

La información de cada evento adverso se obtiene a partir de la recolección de datos por medio de formularios impresos, que en muchos casos se realiza por personas ajenas a CLOPAD no suficientemente capacitadas para ello, esto ocasiona que la información generada sea ambigua, subjetiva y no confiable. La consulta y análisis de datos para decidir las acciones a realizar frente a un desastre se hace difícil y lento, además si se cuenta con la información no apropiada se pueden presentar errores al momento de prevenir, atender o mitigar un desastre.

Teniendo en cuenta lo anterior se propuso crear un sistema que mejore la gestión de información relacionada con los procesos que se llevan a cabo en la evaluación de daños y cubrimiento de necesidades en una población y/o elementos afectados. El desarrollo de la investigación proporciona una herramienta que le permite solo al personal debidamente autorizado realizar la incorporación de información, lo que garantiza que los datos sean correctos y confiables. Adicionalmente estará disponible para personas ajenas al comité sólo para la consulta de información general, a través de reportes e informes.

El sistema de información desarrollado busca mejorar el registro, consulta y análisis de los datos, permitiendo realizar de forma adecuada y oportuna la atención ante un evento catastrófico. Además, mediante la utilización de una aplicación Web permitirá acceder a la información a través de Internet en cualquier lugar, y la centralización de los datos; ésta característica del sistema evitará información redundante, optimizará el tiempo de consulta y minimizará los costos en los que incurre el CLOPAD para el proceso de EDAN como los formularios impresos, tablas de soporte, lapiceros, etc.

Por otro lado, la toma de decisiones oportunas en la atención de desastres, se basa en el análisis de gran cantidad de información disponible que permite identificar las necesidades sobre las que se tiene que actuar, para lo cual es fundamental un manejo apropiado de la información. Teniendo en cuenta este aspecto el sistema facilitará la gestión de información permitiendo generar informes y reportes que ayuden en la toma de decisiones adecuadas y oportunas que respondan a las prioridades de la población afectada.

Finalmente, se optó por aprovechar la disponibilidad del software libre para la realización del proyecto, sumándose el aporte informático para el comité y el área técnica de la alcaldía de la ciudad perteneciente a él, así como los beneficios que puede traer a la comunidad.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Desarrollar un Sistema de Información mediante la construcción de una aplicación Web, para el manejo de información en las actividades relacionadas con la atención de desastres en el CLOPAD de la ciudad de Tuluá.

3.2 OBJETIVOS ESPECÍFICOS

1. Implementar un módulo que permita el manejo de información en la etapa de atención de desastres, llevada a cabo mediante la evaluación de daños y análisis de necesidades.
2. Implementar un módulo que permita el manejo de información sobre los recursos que dispone el CLOPAD para la atención de desastres.
3. Permitir la generación de reportes parciales y/o totales en las etapas de atención y generación de necesidades.
4. Implementar un módulo que utilice un modelo de datos geográfico generado sobre la información de desastres que permita visualizar gráficamente su ubicación y descripción a través de un mapa.
5. Implementar una aplicación web para dispositivos móviles que permita el registro de información más relevante de la etapa de atención de un desastre.
6. Construir una aplicación de respaldo para dispositivos móviles que permita el registro de información de desastres en el dispositivo cliente, en caso de no existir comunicación con la aplicación Web principal.

4. MARCO DE REFERENCIA

4.1 MARCO TEÓRICO

4.1.1 Sistemas de Información

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades relacionadas con el manejo de información de una empresa o negocio [16].

Un sistema de información cuenta con tres elementos principales:

- El equipo computacional: el hardware necesario para que el sistema de información pueda operar.
- El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.
- La información.

4.1.1.1 Actividades

- Entrada de Información

Proceso en el cual se realiza la toma de los datos que requiere el sistema para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

- Almacenamiento de información

El almacenamiento es una de las actividades más importantes que debe establecer el sistema de información, ya que a través de esta propiedad el sistema puede recordar la información guardada, generada en todos y cada uno de los procesos llevados a cabo en él.

- Procesamiento de Información

Esta actividad hace referencia a la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones [16].

- Salida de Información

Capacidad del Sistema de Información de mostrar la información procesada o bien datos de entrada al exterior.

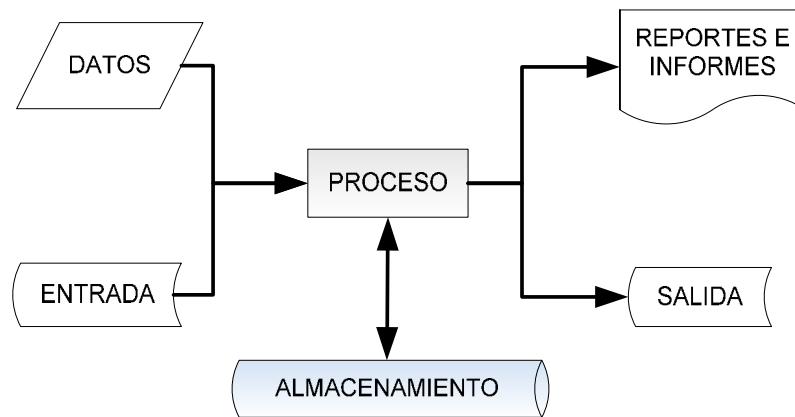


Figura 1. Flujo de actividades de un Sistema de Información

4.1.1.2 Objetivos

- Automatizar los procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

4.1.1.3 Tipos

- Sistemas Transaccionales

Permiten automatizar los procesos operativos de una organización, donde se lleva a cabo el procesamiento de transacciones como son los pagos, cobros, pólizas, entradas, salidas, etc.

Los sistemas transaccionales permiten ahorrar gastos en mano de obra, apoyando las tareas que se realizan en la organización a nivel operativo a través de la construcción de grandes bases de información que posteriormente pueden ser analizadas y útiles en la toma de decisiones.

- Sistemas de Apoyo en la toma de Decisiones

Este tipo de sistema no solo permite el registro y almacenamiento de información, sino que establece un análisis de datos que proporcione a los usuarios las herramientas adecuadas y oportunas para la toma de decisiones en su organización. Suelen introducirse después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información [16]. Un aspecto importante que cabe mencionar sobre estos sistemas tiene que ver con la mano de obra. A diferencia de los sistemas transaccionales, este tipo de sistema no suele ahorrar mano de obra lo que hace difícil el desarrollo de estos sistemas, ya que no funcionan como sistemas expertos y requieren del manejo de personas para evaluar las decisiones que este genere.

- Sistemas Estratégicos

Este tipo de sistemas, se desarrolla en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información. Su función primordial no es apoyar la automatización de procesos operativos ni

proporcionar información para apoyar la toma de decisiones, su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. Este tipo de sistema suele desarrollarse en cada organización y es independiente de cualquier otra competencia, ya que es generado mediante un proceso de evolución dentro de la organización donde se está implementando.

4.1.2 Aplicaciones Web

Una aplicación Web es un desarrollo informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero y a la habilidad para actualizarse y mantenerse sin distribuir e instalar software en miles de potenciales clientes.

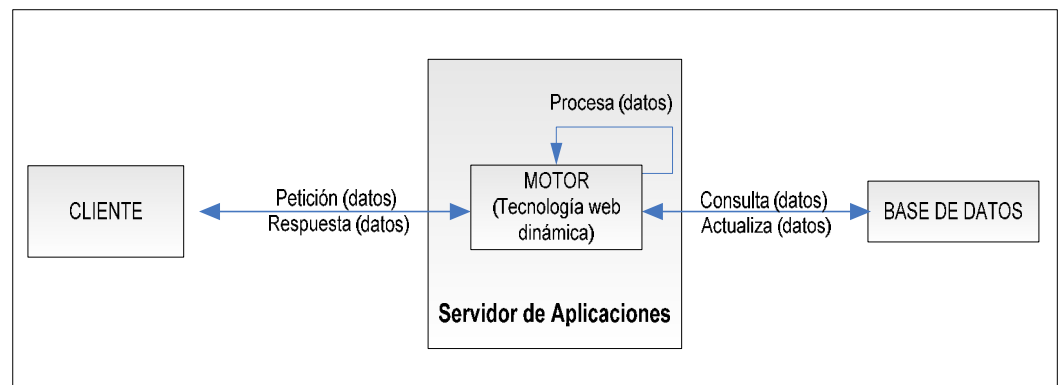


Figura 2. Capas de una aplicación Web

El usuario interactúa con las aplicaciones web a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario. Por tanto, el sistema se distribuye en tres componentes: el navegador, que presenta la interfaz al usuario; la aplicación, que se encarga de realizar las operaciones necesarias según las acciones llevadas a cabo por éste y la base de datos, donde la información relacionada con la aplicación se hace persistente. Esta distribución se conoce como el modelo o arquitectura de tres capas [3]. (Ver Figura 2).

Generalmente, el navegador suele ser solo el componente que despliega información al usuario (modelo de cliente delgado), y no lleva a cabo ningún procesamiento relacionado con la lógica de negocio [3]. Sin embargo, se puede utilizar por ejemplo código javascript, lo que podría situar este componente en un punto intermedio entre un modelo de cliente delgado y un modelo de cliente grueso (donde el cliente realiza el procesamiento de la información y el servidor sólo es responsable de la administración de datos). No obstante, el procesamiento

realizado en el cliente suele estar relacionado con aspectos de la interfaz y nunca con la lógica de negocio. En este sentido, así en el navegador se utilice ciertas herramientas que lo permitan definir como un modelo intermedio, el servidor será quien cumpla la función de procesamiento de información.

Una aplicación web, se puede dividir entonces, según [3] en tres áreas o niveles:

- **Nivel de presentación:** encargado de generar la interfaz de usuario en función de las acciones llevadas a cabo por el mismo.
- **Nivel de negocio:** contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario.
- **Nivel de administración de datos:** encargado de hacer persistente toda la información, suministrar y almacenar información para el nivel de negocio.

4.1.2.1 Patrones de Diseño

- Modelo-Vista-Control

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee datos dinámicos a la página, el modelo es el sistema de clases que representan la abstracción de los elementos del mundo real que intervienen en la aplicación y el controlador representa la lógica de negocio [1].

En la figura siguiente, se ve la arquitectura MVC en su forma más general.

- Una vista envía o recibe datos de un Controlador
- Una vista utiliza los datos de un Modelo
- Un Controlador maneja un Modelo

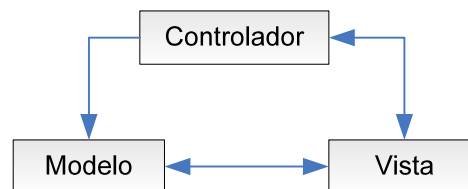


Figura 3. Componentes MVC

Definición de las partes

La Vista es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a ellas cuando cambia el modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

Flujo de datos a través de los componentes de la Arquitectura MVC

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón o un enlace).
2. El controlador recibe, por parte de los objetos de la vista, la notificación de la acción solicitada por el usuario.
3. El controlador gestiona el evento que llega.
4. El controlador accede al modelo, consultándolo o modificándolo de acuerdo a la acción solicitada por el usuario. Los controladores complejos están estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
5. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
6. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

4.1.2.2 Java y J2SE

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Este lenguaje, debido a la explosión y diversidad tecnológica de los últimos años, se ha ido adaptando a las necesidades actuales, tanto de las empresas como de los usuarios, desarrollando soluciones personalizadas para cada ámbito tecnológico. Con este objetivo Sun creó una serie de versiones o ediciones enfocadas a distintos ámbitos o entornos.

En la actualidad se cuenta con tres grandes ediciones:

- J2SE (Java 2 Standard Edition): Orientada al desarrollo de aplicaciones de usuario final. Esta versión estándar fue la que se utilizó para el desarrollo de la aplicación.

- J2EE (Java 2 Enterprise Edition): Orientada al entorno empresarial. Esta edición está orientada especialmente al desarrollo de servicios web, servicios de nombres, persistencia de objetos, XML, autenticación, APIs (Application Program Interface) para la gestión de transacciones, etc. El cometido de esta especificación es ampliar la J2SE para dar soporte a los requisitos de las aplicaciones de empresa.

- J2ME (Java 2 Micro Edition): Orientada a la programación de aplicaciones para pequeños dispositivos electrónicos. Los cuales tienen unas capacidades, computacionales y gráficas, reducidas. Esta versión de Java se verá con más detalle en el tema de aplicaciones móviles.

Características

Java fue diseñado para ser un lenguaje sencillo, orientado a objetos, con alto rendimiento, además es un lenguaje gratuito. Las características más importantes según [9] son:

- Orientada a Objetos

Se refiere a un método de programación y al diseño del lenguaje. Los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el "comportamiento" (el código) y el "estado" (datos). Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es que las aplicaciones sean más fáciles de gestionar y manejar, mejorando como consecuencia su calidad. Otra de las ventajas de este tipo de programación, es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, uno de los objetivos fundamentales de la Ingeniería del Software.

- Independiente de plataforma

Java está diseñado para soportar aplicaciones que pueden ser instaladas en un entorno de red heterogéneo, con hardware y sistemas operativos diversos. Para hacer esto, el compilador Java genera 'bytecodes', un formato de código independiente de la plataforma diseñado para transportar código eficientemente a través de múltiples plataformas de hardware y software. El 'bytecode' es traducido a código máquina y ejecutado por la Java Virtual Machine, que es la implementación Java para cada plataforma hardware-software concreta.

4.1.3 Aplicaciones Móviles

A medida que pasa el tiempo, las capacidades de los dispositivos móviles están creciendo, permitiendo ser usados cada vez más con fines de solucionar problemas en la sociedad o mejorar la forma de incorporación de datos a aplicaciones existentes [11]. Sin embargo, la informática móvil es distinta de la informática para PC, por lo tanto hay que diseñarlos de forma distinta. Una aplicación móvil puede en cierta medida tener las características de una aplicación de escritorio, sin embargo muchas de estas pueden ser restringidas dadas las características de los dispositivos donde funcionan.

Las herramientas o lenguajes que generalmente se utilizan para el desarrollo de este tipo de aplicaciones, se mencionan a continuación.

4.1.3.1 J2ME

J2ME (Java 2 Micro Edition) es la versión de Java orientada a los dispositivos móviles. Esta plataforma ofrece una arquitectura basada en una serie de componentes a partir de los cuales se construye una implementación concreta para un dispositivo determinado, obteniendo así una arquitectura modular que se adapta a las limitaciones y diferencias existentes entre los distintos dispositivos del área a la cual está orientada.

J2ME nace con el objetivo de permitir que aplicaciones Java se ejecuten en dispositivos cuyos recursos son limitados.

Arquitectura

Dadas las características de los dispositivos en los que se ejecutan estas aplicaciones y con el fin de establecer un entorno de ejecución para cualquier tipo de dispositivo se establecen diversos componentes que hacen parte de una arquitectura genérica de desarrollo. Estos componentes se muestran en la figura 4.

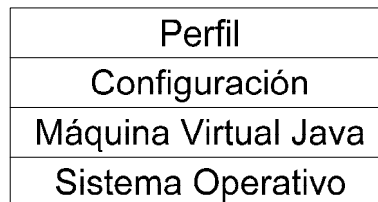


Figura 4. Arquitectura genérica de J2ME

Existen dos tipos de máquina virtual java, cada una para diferentes tipos de dispositivos, estas son: KVM (K Virtual Machine) y CVM (Compact Virtual Machine). En cuanto a la configuración, se refiere a un conjunto de clases básicas orientadas a realizar implementaciones para una familia determinada de dispositivos que tienen ciertas características comunes. Las configuraciones existentes son CDC (Connected Device Configuration) y CLDC (Connected Limited Device Configuration), y por último el perfil que es igualmente un conjunto de clases, pero ya más específicas orientadas a implementar funcionalidades de más alto nivel según la configuración establecida para el dispositivo.

Los programas construidos con J2ME pueden almacenar datos en una memoria de forma persistente, la aplicación puede correr aun si no hay conexión al servidor, lo que indica que puede presentar un servicio de captura y visualización de datos, que serán enviados cuando se establezca conexión con el servidor.

4.1.3.2 WML

Es el lenguaje de Marcado Inalámbrico (WML). Éste describe como es presentado el contenido al dispositivo inalámbrico, permitiendo visualizar información, presentar opciones de entrada, y decir a los agentes de usuario (programas que interpretan WML, WMLscript, y otras formas de codificación- típicamente un mini buscador en un teléfono móvil) cómo responder una vez que una opción ha sido seleccionada [4].

La interfaz de usuario para WML es un mini explorador WAP mejorado para ubicarse en dispositivos inalámbricos móviles. Un documento WML es llamado un *deck*, que es comparable con una página HTML. Al contrario de la monótona estructura del contenido HTML, los documentos WML – o *decks*- son divididos en unidades separadas de interacción del usuario. Cada unidad es llamada una *card*, y los servicios WAP son creados para permitir la navegación del usuario entre *cards* de uno o más *decks*, de forma similar que los hipervínculos son usados dentro y entre documentos HTML [4].

WAP

WAP (Wireless Applications Protocol, Protocolo de Aplicaciones inalámbricas) es una forma de comunicación que permite el envío de datos entre una aplicación móvil. Dicha comunicación se establece vía Internet usando un browser y el protocolo HTTP (HiperText Transfer Protocol, Protocolo de transferencia de hipertexto) [11].

La arquitectura WAP aplica estándares de Internet a tecnología de miniexplorador con el dispositivo inalámbrico que controla cómo se muestra el contenido del servidor WAP. El modelo del protocolo está basado directamente en el modelo familiar de World Wide Web, pero ha sido optimizado para proveer funcionalidades a través de redes inalámbricas entre terminales inalámbricos [4].

WAP usa una arquitectura cliente/servidor que emplea un sencillo minibuscador inalámbrico y requiere simplemente limitados recursos y un gateway WAP para entregar información desde el servidor donde se encuentra guardado. Este es un estándar independiente de la interfaz de usuario, y el soporte de datos principal. Por lo tanto, es completamente interoperable.

El contenido Web está disponible sobre redes de comunicación inalámbrica a través de un gateway WAP. La figura 5 ilustra como una infraestructura WWW, basada en el protocolo de marcado de hipertexto (HTTP), usa un gateway WAP para unirse con las redes inalámbricas traduciendo las peticiones HTTP dentro de peticiones del dispositivo inalámbrico.

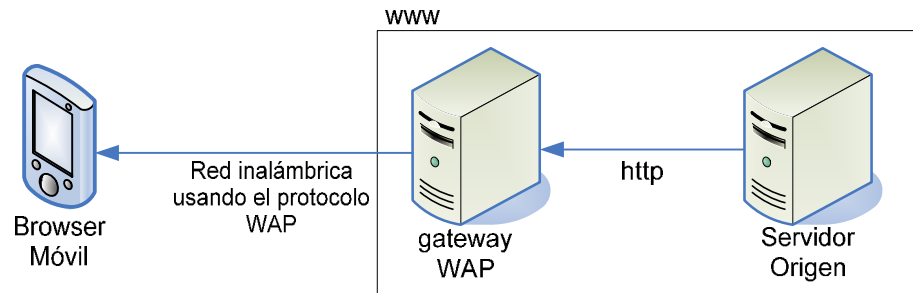


Figura 5. Gateway WAP

4.1.4 Sistema de Información Espacial en la Web

Un sistema de Información espacial, que también recibe el nombre de aplicación cartográfica bajo la web, como cualquier sistema, es una integración organizada de hardware, software además de datos geográficos, diseñado con el fin de capturar, almacenar, manipular, analizar y desplegar información geográficamente referenciada. Dicha información también puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información.

El sistema funciona como una base de datos con información geográfica que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

Un sistema en la web que procese información espacial, tiene 3 componentes esenciales: la geografía o el entorno que se desea modelar, la información que contendrá la representación de esos elementos de la geografía y la Web como medio de visualización de dicha información.

4.1.4.1 Funciones

En la actualidad la web, se ha convertido en un elemento primordial de infraestructura tecnológica para adquirir y distribuir información, esto inevitablemente ha generado un aumento de los intereses a nivel corporativo, especialmente por parte de las instancias gubernamentales y de la administración. De acuerdo a ello se establecen diversas funciones que un sistema de este tipo debe proporcionar:

- **Localización:** preguntar por las características de un lugar concreto.
- **Condición:** el cumplimiento o no de unas condiciones impuestas al sistema.
- **Tendencia:** comparación entre situaciones temporales o espaciales distintas de alguna característica.
- **Rutas:** cálculo de rutas óptimas entre dos o más puntos.
- **Pautas:** detección de pautas espaciales.
- **Modelos:** generación de modelos a partir de fenómenos o actuaciones simuladas.

4.1.4.2 Arquitectura

Una aplicación cartográfica en la web como cualquier aplicación que corra en internet, está basada en el modelo cliente servidor o estructura de tres capas como se menciono anteriormente en el tema de aplicaciones web. La única diferencia radica en la utilización de un servidor de mapas que permita el despliegue de información geográfica. Además, en que las solicitudes al servidor son correspondientes a consultas o registro de información espacial sobre un mapa. La estructura expuesta se muestra en la siguiente figura, donde el cliente hace una solicitud al servidor, el cual la procesa y retorna la información al cliente.

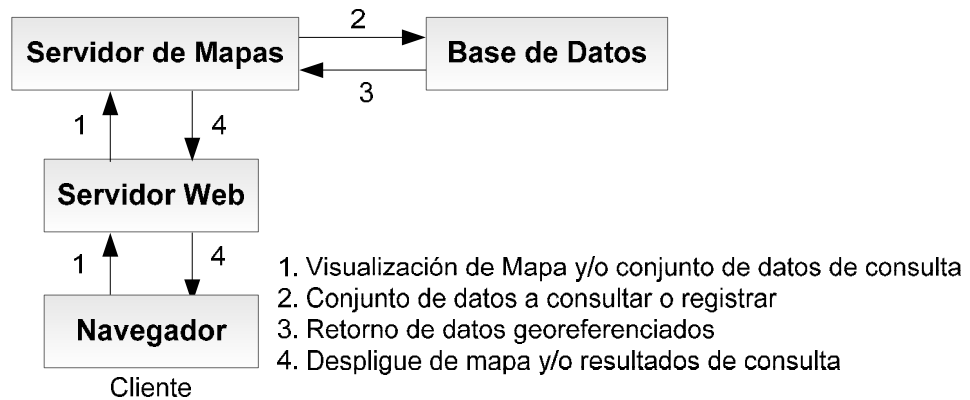


Figura 6. Modelo básico de funcionamiento de una aplicación cartográfica en la web

Arquitectura de cliente delgado

En esta arquitectura, los objetos visuales requeridos por el cliente son enviados en formatos raster tales como jpg, gif, etc. La mayoría de los análisis geográficos son manejados por el servidor, mientras el cliente solo visualiza páginas HTML y ejecuta código javascript. En esta arquitectura los datos requeridos por un cliente son preparados en el servidor, el cual llama al servidor de mapas para que mediante un CGI y/o alguna API procese la solicitud el cliente, comunicándose a la base de datos o aplicación relacionada, devolviendo una respuesta al servidor web que se encarga de enviarla al browser del cliente en formato HTML (Ver figura 7) [2].

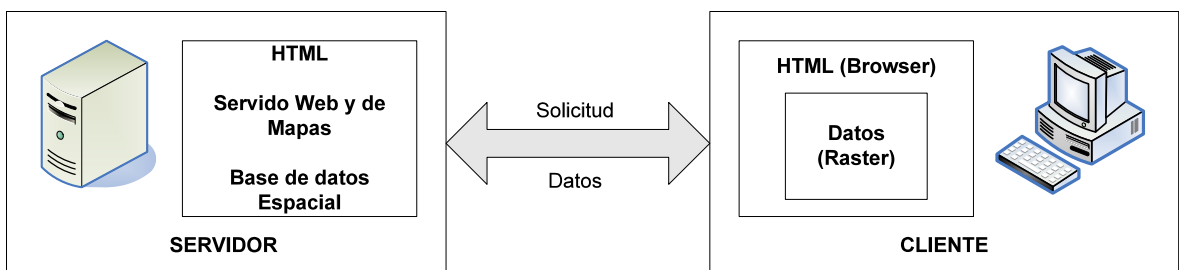


Figura 7. Arquitectura de cliente delgado

Arquitectura de cliente grueso

Aquí las tareas del servidor son limitadas a solo el envío de datos, mientras la mayoría del análisis geográfico es manejado por la plataforma del cliente. El browser del cliente es mejorado agregándole nuevos plug-ins como el lenguaje java, y applet. (Ver figura 8).

El servidor envía al cliente los datos en formato vector, y este se encarga del procesamiento, y visualización de la información, reduciendo la carga en el servidor [2].

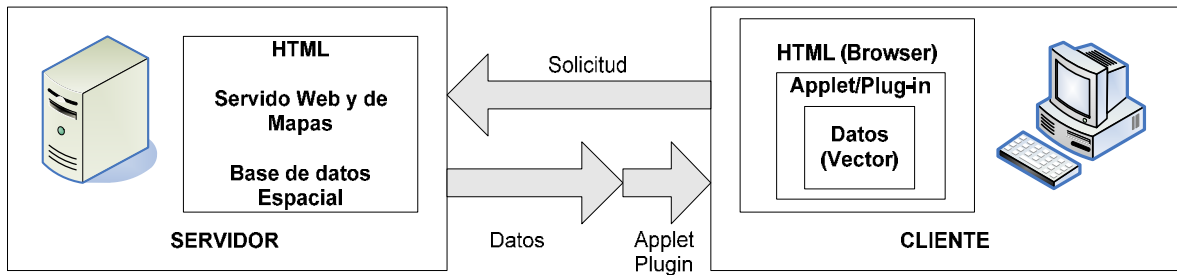


Figura 8. Arquitectura de cliente grueso

4.1.4.3 Bases de datos Espaciales

Es un sistema administrador de bases de datos que maneja datos existentes en un espacio o *datos espaciales*. El espacio establece un marco de referencia para definir la localización y relación entre objetos.

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que pueda ser procesada por el lenguaje de las computadoras actuales. Este proceso de abstracción tiene diversos niveles y normalmente comienza con la concepción de la estructura de la base de datos, generalmente en capas; en esta fase, y dependiendo de la utilidad que se vaya a dar a la información a compilar, se seleccionan las capas temáticas a incluir.

La estructuración de la información espacial procedente del mundo real en capas conlleva cierto nivel de dificultad. En primer lugar, la necesidad de abstracción que requieren los computadores implica trabajar con primitivas básicas de dibujo, de tal forma que toda la complejidad de la realidad ha de ser reducida a puntos, líneas o polígonos. En segundo lugar, existen relaciones espaciales entre los objetos geográficos que el sistema no puede obviar; la topología, que en realidad es el método matemático-lógico usado para definir las relaciones espaciales entre los objetos geográficos, puede llegar a ser muy compleja, ya que son muchos los elementos que interactúan sobre cada aspecto de la realidad.

La información puede ser almacenada en formato raster o vectorial. El modelo de SIG raster o de retícula se centra en las propiedades del espacio más que en la precisión de la localización. Divide el espacio en celdas regulares donde cada una de ellas representa un único valor. Cuanto mayor sean las dimensiones de las celdas (resolución) menor es la precisión o detalle en la representación del espacio geográfico. En el caso del modelo de SIG vectorial, el interés de las representaciones se centra en la precisión de localización de los elementos sobre

el espacio y donde los fenómenos a representar son discretos, es decir, de límites definidos. Para modelar digitalmente las entidades del mundo real se utilizan tres objetos espaciales: el punto, la línea y el polígono.

Los SIG vectoriales son más populares en el mercado. No obstante, los SIG raster son muy utilizados en estudios que requieran la generación de capas continuas, necesarias en fenómenos no discretos; también en estudios medioambientales donde no se requiere una excesiva precisión espacial (contaminación atmosférica, distribución de temperaturas, localización de especies marinas, análisis geológicos, etc.).

Tipos de datos espaciales

Un modelo de datos geográfico es una abstracción del mundo real que emplea un conjunto de objetos dato, para soportar el despliegue de mapas, consultas, edición y análisis. Los datos geográficos, presentan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, redes, superficies, ubicaciones e imágenes, a los cuales se les asignan sus respectivos atributos que los definen y describen [10].

Un dato espacial es una variable asociada a una localización del espacio. Normalmente se utilizan datos vectoriales, los cuales pueden ser expresados mediante tres estructuras:

- Puntos

Se encuentran determinados por las coordenadas terrestres medidas por latitud y longitud. Por ejemplo, ciudades y accidentes geográficos puntuales.

- Líneas

Objetos abiertos que cubren una distancia dada y comunican varios puntos o nodos, aunque debido a la forma esférica de la tierra también se le consideran como arcos. Líneas telefónicas, carreteras y vías de trenes son ejemplos de líneas geográficas.

- Polígonos

Figuras planas conectadas por distintas líneas u objetos cerrados que cubren un área determinada, como por ejemplo países, regiones o lagos.

La información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas (x, y). La ubicación de una característica puntual, pueden describirse con un sólo punto (x, y). Las características lineales, pueden almacenarse como un conjunto de puntos de coordenadas (x, y). Las características poligonales, pueden almacenarse como un circuito cerrado de coordenadas.

Los datos espaciales además se caracterizan por su naturaleza geo-referenciada y multidireccional. La primera se refiere a que la posición relativa o absoluta de cualquier elemento sobre el espacio contiene información valiosa, pues la localización debe considerarse explícitamente en cualquier análisis. Por multidireccional se entiende a que existen relaciones complejas no lineales, es decir que un elemento cualquiera se relaciona con su vecino y además con regiones lejanas, por lo que la relación entre todos los elementos no es unidireccional. Es decir, todos los elementos se relacionan entre sí, pero existe una relación más profunda entre los elementos más cercanos.

4.2 MARCO CONTEXTUAL

La función principal del sistema desarrollado es ayudar en la atención de desastres a nivel local. El sistema está dirigido principalmente a los responsables de la toma de decisiones en el Comité Local de Prevención y Atención de Desastres de la ciudad de Tuluá que manejan información en cada una de las etapas que se presentan en la ocurrencia de un desastre: antes, durante y después. Para ello, el sistema proporciona información completa y organizada de la etapa de atención de desastres, almacenada en forma de documentos, registros, informes o reportes que faciliten la consulta y el análisis de ella.

Para mantener actualizados los datos el sistema cuenta con usuarios que nutren la base de datos oportunamente y con los datos adecuados obtenidos de las diversas fuentes existentes (organizaciones, personas, investigaciones), dicho manejo de administradores evitará que el sistema albergue información incorrecta y/o incompleta generada por personas no altamente capacitadas o autorizadas para el registro de información.

El sistema de información tipo web permite el manejo de los aspectos fundamentales en la evaluación de daños y análisis de necesidades, procesos de la etapa de atención donde se incluye información sobre el evento y se establecen los recursos necesarios para atender a la población.

El sistema desarrollado proporciona una herramienta muy eficaz que facilita todo el manejo de información que se requiere en actividades tan importantes como son la atención de desastres, sin embargo el desarrollo de este trabajo es un poco limitado ya que se maneja solo a nivel local como se menciono anteriormente.

Algunos de los sistemas que ya han sido desarrollados en Colombia o en países cercanos similares al planteado en este documento son los siguientes:

- SINPAD (Sistema de Información de Prevención y Atención de Desastres-Perú): opera en el Centro Operativo de Emergencias Nacionales del INDECI en Lima. Su función principal es la de atención de emergencias en Perú mediante el registro y seguimiento de eventos notificados sobre peligros inminentes y eventos notificados a nivel nacional.
- CENAPRAD es el Centro Nacional de Prevención y Atención ante Desastres de la Dirección Nacional de Protección Civil y Administración de Desastres (Ministerio de Interiores y Justicia) en Venezuela: El sistema tiene como misión prevenir y alertar ante amenazas naturales y antrópicas para contribuir en reducir el riesgo de ocurrencia de desastre y a la vez ejercer control de las operaciones durante la existencia de un evento.
- DESINVENTAR que incluye un producto de software con dos componentes principales, el módulo DesInventar, a través del cual se alimenta la base de datos, mediante campos predefinidos (información espacio temporal, tipos de eventos y causas, fuentes), efectos directos e indirectos (sobre la vida, viviendas, infraestructura, sectores económicos) y el módulo DesConsultar que permite acceder a la base de datos mediante consultas que relacionen las diversas variables (efectos, eventos, lugares, fechas, etc.), así como su representación mediante tablas de consultas, gráficos y cartografía temática.

4.3 MARCO CONCEPTUAL

- **ActionListener**
Es una clase Interface de Java que tiene un sólo método: void actionPerformed(ActionEvent e). Se usa para detectar y manejar eventos de acción que tienen lugar cuando se produce una acción sobre un elemento del programa. Un evento ActionEvent se produce:
al pulsar un botón (Button)
al hacer doble clic en un elemento de lista (List)
al pulsar INTRO en una caja de texto (TextFiel)
al elegir un menú (MenuItem)
- **Ajax**
Asynchronous JavaScript And XML. es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la aplicación.
- **Amenaza**
Factor externo de riesgo, representado por la posibilidad de que ocurra un fenómeno o un evento adverso que podría generar daño en las personas o su entorno, derivado de la naturaleza, de la actividad humana o de una combinación de ambos, y que puede manifestarse en un momento y un lugar específicos con una magnitud determinada.
- **Apache**
Es un servidor Web HTTP de código abierto para plataformas Unix, Windows, Macintosh.
La arquitectura del servidor Apache es muy modular. El servidor consta de una sección *core* y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web.
- **CVM**
Connected Limited Device Configuration. Es una especificación de un marco de desarrollo para aplicaciones codificadas en J2ME ejecutadas en dispositivos con recursos computacionales muy limitados tales como teléfonos móviles.
- **CLOPAD**
Comité Local para la Prevención y Atención de Desastres. Conjunto de entidades Públicas, Privadas y Comunitarias Integradas, con el objeto de dar solución a los problemas de seguridad de la población que presenten en su entorno físico por la eventual ocurrencia de fenómenos naturales o antrópicos.

- **Desastre**
Alteraciones intensas en las personas, los bienes, los servicios y el medio ambiente, causadas por un suceso natural o generado por la actividad humana, que exceden la capacidad de respuesta de la comunidad afectada.
- **EDAN**
Evaluación de Daños y Análisis de Necesidades. Procesos llevados a cabo en eventos de desastre con el fin de identificar las pérdidas o problemas generados y establecer los recursos necesarios a utilizar para la atención de estos.
- **Evento Adverso**
Fenómeno que produce cambios desfavorables en las personas, la economía, los sistemas sociales o el medio ambiente; puede ser de origen natural, generado por la actividad humana o de origen mixto y puede causar una emergencia o un desastre.
- **HTML**
(*HyperText Markup Language*- Lenguaje de Marcado de Hipertexto) Lenguaje en el que se escriben las páginas web a las que se accede a través de navegadores WWW. Es una aplicación de la ISO Standard 8879:1986.
- **Internet**
Nombre que se le da al conjunto mundial de redes y enrutadores conectados mediante diferentes medios como cables telefónicos, redes de fibra óptica, microondas, antenas parabólicas, satélites y hasta cables submarinos, que utilizan el protocolo TCP/IP para comunicarse entre sí.
- **J2ME**
La plataforma J2ME es una familia de especificaciones que definen varias versiones minimizadas de la plataforma Java 2; estas versiones minimizadas pueden ser usadas para programar en dispositivos electrónicos; desde teléfonos celulares, en PDAs, hasta en tarjetas inteligentes, etc. Estos dispositivos presentan en común que no disponen de abundante memoria ni mucha potencia en el procesamiento [15].
- **JavaScript**
Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.
JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.
- **JVM**
Java Virtual Machine. Una Máquina virtual Java es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), que es

generado por el compilador del lenguaje Java. El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico. La JVM es una de las piezas fundamentales de la plataforma Java, básicamente se sitúa en un nivel superior al Hardware del sistema sobre el que se pretende ejecutar la aplicación, y éste actúa como un puente que entiende tanto el bytecode, como el sistema sobre el que se pretende ejecutar. Así, cuando se escribe una aplicación Java, se hace pensando que será ejecutada en una máquina virtual Java en concreto, siendo ésta la que en última instancia convierte de código bytecode a código nativo del dispositivo final.

- **J2SE**
Java 2 Platform, Standard Edition. Un kit de desarrollo de software que se utiliza para construir aplicaciones con el lenguaje de programación Java.
- **JSP**
Java Server Pages. Es una tecnología Java que permite a los programadores generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento web.
- **KVM**
kilobyte Virtual Machine. Es un ambiente de ejecución de Java completo para dispositivos móviles. Contiene un subconjunto de las clases de Java incluso las necesarias para el manejo de interfaces gráficas.
- **Mapfile**
Es un archivo de configuración que define los datos a ser usados en una aplicación SIG. El Mapfile también contiene información acerca de cómo se debe dibujar el mapa, la leyenda y el resultado de realizar una consulta. El Mapfile tiene normalmente una extensión .map.
- **MapServer**
Es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG en Internet e Intranet con el fin de visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet Map Server (IMS) [17] [8].
- **MapScript**
Es un modulo cargable dinámicamente que hace a las funciones y clases de MapScript de Mapserver disponibles en el ambiente de desarrollo PHP [14].
- **MIDlets**
Es una programa Java para dispositivos móviles, más específicamente para la Máquina Virtual Java MicroEdition (Java ME). Generalmente los Midlets son juegos y aplicaciones que corren en un teléfono celular.

- **PHP**
PHP Hypertext Pre-processor. PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor.
 PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno.
- **PostGis**
 Es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL para su utilización en Sistema de Información Geográfica [7]. Se publica bajo la Licencia pública general de GNU. PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL, y a juicio de muchos, es muy similar a la versión geográfica de Oracle [13].
- **PostgreSQL**
 Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. Como muchos otros proyectos de código libre, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).
- **Protocolo TCP/IP**
 El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN).
- **RecordStore**
 Una clase perteneciente a la versión J2ME de Java que representa a un almacén de registro. Un almacén de registro consta de una colección de registros los cuales se recordarán persistentemente a través de múltiples invocaciones del MIDlet. La plataforma del dispositivo donde se almacena, es la encargada de hacer su mejor esfuerzo para mantener la integridad de los almacenes de registro del MIDlet a lo largo del uso normal de la plataforma, incluyendo reinicio, cambios de batería, etc.
- **Red**
 Conjunto de dispositivos, mecanismos y procedimientos asociados que están conectados por medio de recurso de comunicaciones para intercambiar información útil.
- **Servidor web**
 Es un programa que implementa el protocolo HTTP (*hypertext transfer protocol*). Este protocolo está diseñado para transferir lo que se conoce como hipertextos,

páginas Web o páginas HTML (*hypertext markup language*) que son textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP hechas por un cliente que se conoce como no es más que un navegador. Este realiza una petición al servidor que le responde con el contenido que el cliente solicita. El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y demás objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

- **Servlet**

Es un pequeño programa de Java que se ejecuta con un servidor Web. Recibe y responde a peticiones de clientes Web, usualmente a través de HTTP. Para implementar esta clase se debe escribir un Servlet genérico que hereda de `javax.servlet.GenericServlet` o escribir un Servlet de HTTP que hereda de `javax.servlet.http.HttpServlet`.

- **Shapefile**

Es un formato vectorial de almacenamiento digital donde se guarda la localización de los elementos geográficos y los atributos asociados a ellos.

- **Tomcat**

Es un servidor Web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

- **UTM**

Universal Transverse Mercator. El sistema de coordenadas es un método usado para identificar localizaciones en la tierra pero difiere del método tradicional de latitud y longitud en varios aspectos.

- **Vulnerabilidad**

Susceptibilidad o la predisposición intrínseca de un elemento o de un sistema de ser afectado gravemente. Es el factor interno del riesgo, debido a que esta situación depende de la actividad humana.

- **WAP**

Wireless Application Protocol (protocolo de aplicaciones inalámbricas) es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, como el acceso a servicios de Internet desde un dispositivo móvil.

Se trata de la especificación de un entorno de aplicación y de un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos, se pueden utilizar para acceder a diversas aplicaciones de internet.

- **WML**

Wireless Markup Language. Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología WAP.

La visualización de la página dependerá del dispositivo que se use y de la forma en que este interprete el código, ya que varían entre si. WML es un metalenguaje, lo que implica que además de usar etiquetas predefinidas se pueden crear componentes propios.

- **WWW**

Es un sistema hipermedia interactivo desarrollado sobre Internet. Su función es unir texto, imágenes, audio y vídeo dentro de un mismo envoltorio llamado documento. WWW se asienta sobre el protocolo HTTP y sobre el lenguaje de definición de documentos hipermedia HTML.

Sistema de información global, interactivo, dinámico, de plataforma independiente; basado en hipertextos que corre sobre Internet.

5. PROCESO DE DESARROLLO

La metodología general de trabajo se enfocó en 2 fases, las cuales se definen como:

Investigación sobre los procesos de CLOPAD

Esta fase consistió en la realización de entrevistas a miembros del CLOPAD para conocer claramente los procesos que ellos manejan e identificar las necesidades y problemas que se presentan en estos. Además se llevaron a cabo procesos de investigación sobre los principales procesos de toma de decisiones que se llevan a cabo en el Comité y la determinación del flujo de información.

Diseño e Implementación del Sistema de Información Web

En esta fase se definieron claramente las tecnologías y/o herramientas requeridas para el desarrollo de la aplicación Web y se investigaron además las que permitieran implementar las características adicionales definidas para el sistema:

- Utilización de la aplicación en dispositivos móviles.
- Visualización de la ubicación y descripción de desastres a través de mapas.
- Aplicación de respaldo en móvil para el registro y almacenamiento de información de desastres.

Para el desarrollo de la aplicación en general se utilizó la metodología que corresponde al proceso de Ingeniería Web que está compuesto por siete pasos: formulación, planeación, modelo de análisis, modelo de diseño, construcción, entrega y retroalimentación [5].

5.1 ETAPAS DE DESARROLLO

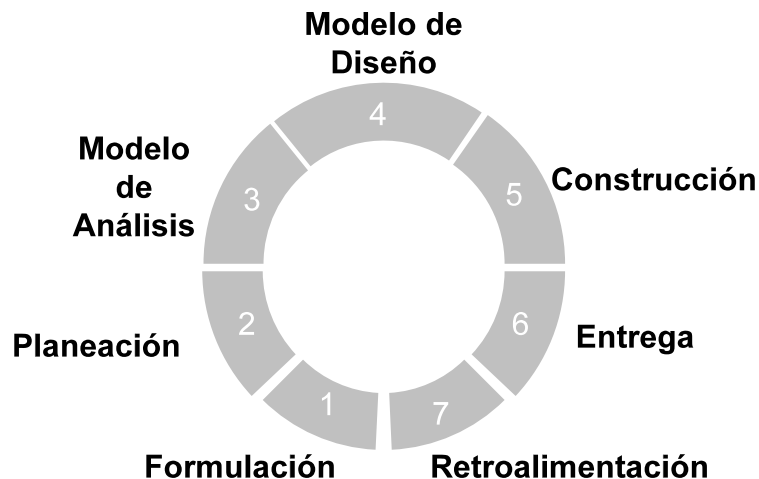


Figura 9. Etapas de desarrollo – Proceso de Ingeniería Web

Formulación

Incluye el levantamiento de información con los usuarios funcionales, con el fin de establecer las necesidades presentadas y así realizar la especificación de requerimientos. Esta etapa se compuso de actividades como la formulación de preguntas, la recolección de requerimientos, el análisis de la información junto con los facilitadores, y los productos a obtener. [6].

Este paso se llevo a cabo en la fase de investigación sobre los procesos del CLOPAD, donde se establecieron las necesidades concretas y las soluciones adoptadas para suplirlas, proceso que facilito el diseño del sistema y la adecuación de los procesos que se manejaban hasta el momento. Las características del sistema actual y de la aplicación planteada se mencionaron anteriormente en los puntos 2 y 4.

Planeación

Se realiza la estimación del costo general del proyecto, la evaluación de riesgos, y se establece el calendario de desarrollo (fechas de entrega).

Modelo de Análisis

Se lleva a cabo el análisis de requerimientos (detallados), y la investigación sobre el problema. Realizando además los diagramas de casos de uso y de secuencia.

Modelo de Diseño

En la etapa de diseño se realizan diferentes actividades que incluyen aplicación de metodologías o técnicas que permitan realizar la documentación del sistema. Además, en esta etapa se realiza el diseño funcional del sistema que incluye la construcción de diagramas que indican su funcionalidad, componentes y relaciones entre estos.

Igualmente, se lleva a cabo la elaboración de prototipos, que da claridad al usuario como al desarrollador de las características con que contará la aplicación.

En este punto se utilizará UML (*Unified Modeling Language*), como lenguaje de ayuda en la especificación, construcción, visualización y documentación de los modelos para sistemas de software.

Construcción

Es la transformación de las especificaciones del diseño en un producto o en una operación. Esta transformación incluye la producción de código en el lenguaje que se haya establecido para implementar la aplicación.

Entrega

En esta etapa se incluye la adecuación del sistema en un ambiente real y la utilización por el usuario final.


Retroalimentación

Evaluación del sistema con los usuarios concretos de la aplicación, con el fin llevar a cabo las pruebas de aceptación y realizar los cambios necesarios.

6. MODELO DE ANÁLISIS

6.1 DEFINICIÓN DE REQUERIMIENTOS

6.1.1 Requerimientos Funcionales

	Especificación de Requerimientos ISO-9000 Universidad del Valle	Documento: ER-001	Revisión: 00 Fecha: 12/09/07
Título: Especificación de Requerimientos Funcionales para el desarrollo del sistema de información web para el manejo de información en el CLOPAD - SIWEPAD			

Funciones – Aplicación Web general

Req #	REQUERIMIENTOS FUNCIONALES	CAT
Usuarios		
1	El sistema deberá ofrecer al usuario una opción que le permita autenticarse e ingresar a la aplicación.	Visible
2	El sistema en el momento en que un usuario se autentique, deberá validar si el usuario existe o no en el sistema.	Oculto
3	El sistema en el momento en que se autentique un usuario, deberá consultar el perfil que éste posee y de acuerdo a él generar el menú que le corresponde.	Oculto
4	El sistema debe permitir al administrador, el registro de usuarios a la aplicación, especificando por cada uno la información general y el perfil.	Visible
5	El sistema debe permitir al administrador, la modificación de usuarios de la aplicación, sea de su información general o de su perfil.	Visible
6	El sistema debe permitir al administrador, la consulta de usuarios de la aplicación de acuerdo a diferentes parámetros de búsqueda, como identificación, nombre o perfil.	Visible
7	El sistema debe permitir al administrador, eliminar usuarios de la aplicación en el momento que éste lo requiera.	Visible
8	El sistema debe permitir al administrador, activar usuarios de la aplicación que él haya eliminado, en el momento que lo requiera.	Visible
9	El sistema debe permitir al administrador registrar nuevos perfiles indicando para éste las acciones o tareas que le serán permitidas.	Visible
10	En el momento en que el administrador lo requiera el	Visible

	sistema debe ofrecerle una opción donde pueda consultar los perfiles que haya registrado en el sistema.	
11	El sistema debe ofrecer una opción que le permita al administrador modificar las acciones que le pertenecen a un perfil.	Visible
12	El sistema debe ofrecer una opción que permita al administrador eliminar un perfil.	Visible
13	El sistema debe ofrecer al Administrador una opción que le permita registrar un grupo de Usuarios, indicando sus datos generales y los integrantes que le pertenecen.	Visible
14	El sistema deberá ofrecer al Administrador una opción que le permita consultar los grupos de Usuarios que se encuentran en el sistema mostrando por cada uno la fecha de registro, el nombre y sus integrantes.	Visible
15	El sistema debe permitir al administrador modificar y eliminar un grupo de usuarios.	Visible
16	En cada una de las actividades de registro del sistema, éste deberá registrar adicional a la información suministrada, el usuario que realizo el registro, la fecha y la hora.	Oculto
Desastres		
17	El sistema debe permitir al usuario, registrar la información general del desastre generado: zona afectada y tipo de evento, fecha y hora de ocurrencia e impacto.	Visible
18	El sistema debe permitir consultar desastres de acuerdo al tipo de evento, fecha de ocurrencia o zona.	Visible
19	En el momento en que el usuario lo requiera el sistema debe ofrecerle una opción que le permita actualizar la información de un desastre.	Visible
20	El sistema debe permitir al usuario, una vez haya consultado un desastre, especificar su ubicación en el mapa de Tuluá, señalando el punto donde desea situarlo.	Visible
21	En el momento que un usuario consulte un desastre, el sistema debe ofrecerle una opción que le permita registrar viviendas afectadas por el evento, especificando su información general como la dirección, el teléfono, el estrato, el formato de tenencia de la vivienda y si es rural o urbana; el estado de sus servicios públicos y daños ocasionados en su infraestructura.	Visible
22	El sistema debe permitir al usuario en el momento en que registre una vivienda especificar las necesidades que esta posee, especificando cada tipo de necesidad y la cantidad de recursos que requiere para suplirla.	Visible
23	El sistema debe permitir al usuario registrar la evaluación agropecuaria y de cultivos de una vivienda, especificando el tipo de cultivos y las pérdidas	Visible

	económicas y materiales.	
24	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las viviendas afectadas por un desastre, visualizando por cada una su información básica.	Visible
25	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las necesidades de una vivienda.	Visible
26	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar la evaluación agropecuaria de una vivienda.	Visible
27	El sistema debe permitir al usuario actualizar la información de una vivienda.	Visible
28	El sistema debe permitir al usuario actualizar la información de las necesidades de una vivienda.	Visible
29	El sistema debe permitir al usuario actualizar la información de la evaluación agropecuaria de una vivienda.	Visible
30	En el momento en un usuario registre una vivienda, el sistema debe ofrecerle una opción que le permita registrar la familia que habita en ella, especificando su información general como sus apellidos, jefe de familia y tipo de comunidad.	Visible
31	El sistema debe permitirle al usuario, una vez registre una familia, especificar las necesidades que ella posee, indicando que necesidades son y la cantidad de recursos que requiere para suplirlas.	Visible
32	En el momento en que se haya registrado una familia, el sistema debe ofrecerle al usuario una opción que le permita registrar las personas que pertenecen a ella, especificando por cada una: su identificación, nombre, apellido, edad, sexo, ocupación, el parentesco con el jefe de la familia, además de su estado de salud y los daños o enfermedades que le haya ocasionado el desastre.	Visible
33	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las familias que pertenecen a una vivienda, visualizando por cada una su información básica.	Visible
34	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las necesidades de una familia.	Visible
35	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las personas que pertenecen a una familia.	Visible
36	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las personas afectadas por un desastre.	
37	El sistema debe ofrecer una opción que le permita al	Visible

	usuario ver la información de cada persona afectada por un desastre.	
38	El sistema debe permitir al usuario actualizar la información de una familia.	Visible
39	El sistema debe permitir al usuario actualizar la información de las necesidades de una familia.	Visible
40	El sistema debe permitir al usuario actualizar la información de las personas que pertenecen a una familia.	Visible
41	En el momento en que el usuario lo requiera el sistema debe permitir registrar la información de la evaluación preliminar de un desastre, especificando las vías de acceso al desastre, las condiciones climáticas, los efectos secundarios que provoco el evento, el estado general de los servicios públicos y de la infraestructura productiva.	Visible
42	El sistema debe permitir al usuario consultar la información de la evaluación preliminar de un desastre.	Visible
43	El sistema debe permitir al usuario actualizar la información de la evaluación preliminar de un desastre.	Visible
44	En el momento en que el usuario lo requiera el sistema debe permitir registrar la información de la evaluación complementaria de un desastre, especificando los efectos secundarios en salud provocados, el estado de cada una de las líneas de servicios públicos y la evaluación general de los daños de la infraestructura productiva en cada una de las actividades de los sectores agropecuario e industrial y manufacturero especialmente.	Visible
45	El sistema debe permitir al usuario consultar la información de la evaluación complementaria de un desastre.	Visible
46	El sistema debe permitir al usuario actualizar la información de la evaluación complementaria de un desastre.	Visible
47	En el momento en que el usuario lo requiera, el sistema debe ofrecerle una opción que le permita consultar la información de la evaluación final de un desastre.	Visible
48	En el momento en que un usuario desee consultar la evaluación final de un desastre, el sistema automáticamente deberá generar la evaluación, especificando la cantidad de viviendas afectadas según el estado, agrupadas por zonas rural y urbana, el total de viviendas afectadas por cada estado; el recurso humano afectado, especificando la cantidad de heridos y muertos; el total de pacientes atendidos a nivel pre hospitalario discriminados por edad y sexo; el total de pacientes atendidos a nivel asistencial agrupados por	Oculto

	edad y sexo y las pérdidas económicas por cada sector afectado en el desastre.	
49	El sistema debe ofrecer una opción que le permita al usuario descargar la evaluación final del desastre en formato pdf.	Visible
50	En el momento en que el usuario desee descargar la evaluación final del desastre, el sistema automáticamente debe generar éste reporte en formato pdf.	Oculto
51	En el momento en que el usuario lo requiera, el sistema debe ofrecerle una opción que le permita visualizar el consolidado de necesidades de las viviendas afectadas por un desastre.	Visible
52	En el momento en que el usuario lo requiera, el sistema debe ofrecerle una opción que le permita visualizar el consolidado de necesidades de las familias afectadas por un desastre.	Visible
53	El sistema debe ofrecer una opción que le permita al usuario descargar el informe de necesidades de las viviendas en formato pdf.	Visible
54	El sistema debe ofrecer una opción que le permita al usuario descargar el informe de necesidades de las familias en formato pdf.	Visible
55	En el momento en que el usuario desee guardar el informe de necesidades de viviendas, el sistema automáticamente debe generar un informe en formato pdf, donde se indique las necesidades generadas y la cantidad de recursos que son requeridos por cada una de ellas.	Oculto
56	En el momento en que el usuario desee guardar el informe de necesidades de familia, el sistema automáticamente debe generar un informe en formato pdf, donde se indique las necesidades generadas y la cantidad de recursos que son requeridos por cada una de ellas.	Oculto
Recursos		
57	El sistema deberá permitir al administrador registrar los recursos necesarios para la atención de los desastres, tal como cantidad de carpas, módulos de vivienda, frazadas, camas plegables, ropa para adultos y para niños, especificando su nombre, la cantidad con la que disponen, la entidad que ofreció los recursos y las necesidades que éste puede suplir.	Visible
58	EL sistema debe ofrecer una opción que le permita consultar la información de los recursos registrados.	Visible
59	EL sistema debe ofrecer una opción que le permita actualizar la información de los recursos del sistema.	Visible
60	EL sistema debe ofrecer una opción que le permita	Visible

	eliminar la información de los recursos.	
Reportes		
61	El sistema en el momento en que el usuario lo requiera, deberá permitirle generar y guardar un reporte general o resumen con la información general del desastre ocurrido.	Visible
62	El sistema debe permitir generar y guardar reportes de la evaluación preliminar de un desastre.	Visible
63	El sistema debe permitir generar y guardar reportes de la evaluación complementaria de un desastre.	Visible
64	El sistema debe permitir generar y guardar reportes de la evaluación final de un desastre.	Visible
65	El sistema debe permitir generar y guardar reportes de las necesidades de las viviendas de un desastre.	Visible
66	El sistema debe permitir generar y guardar reportes de las necesidades de las familias de un desastre.	Visible
Configuración		
67	El sistema debe ofrecer al usuario una opción que le permita registrar noticias, eventos o capacitaciones, especificando la información general e indicando si será publicado en la página principal de la aplicación.	Visible
68	En el momento que se requiera el sistema debe ofrecer una opción que permita consultar las noticias o actividades registradas, visualizándolas por orden de registro.	Visible
69	El sistema debe ofrecer al usuario una opción que le permita modificar el contenido de una noticia o actividad.	Visible
70	El sistema debe ofrecer una opción que le permita al usuario cambiar el estado de publicación de una noticia o actividad registrada.	Visible
71	El sistema debe permitir al usuario eliminar una noticia o actividad.	Visible
72	El sistema debe ofrecer al usuario una opción que le permita registrar un nuevo tipo de necesidad, especificando la información general de ésta e indicando que tipo es, si es de vivienda o de familia.	Visible
73	En el momento que se requiera el sistema debe ofrecer una opción que permita consultar las necesidades registradas, visualizándolas por orden de registro.	Visible
74	El sistema debe ofrecer al usuario una opción que le permita modificar la información de una necesidad.	Visible
75	El sistema debe ofrecer una opción que le permita al usuario cambiar el estado de utilización de la necesidad en el registro de viviendas o familias cuando ésta no desee ser evaluada en estos registros.	Visible
76	El sistema debe permitir al usuario eliminar una necesidad.	Visible
77	El sistema debe ofrecer al usuario una opción que le	Visible

	permita registrar una entidad que pertenezca o tenga relación con el CLOPAD, especificando la información general como la dirección, el teléfono, el coordinador y sus funciones.	
78	En el momento que se requiera el sistema debe ofrecer una opción que permita consultar las entidades registradas.	Visible
79	El sistema debe ofrecer al usuario una opción que le permita modificar la información de una entidad.	Visible
80	El sistema debe permitir al usuario eliminar una entidad.	Visible
81	El sistema debe ofrecer al usuario una opción que le permita registrar un nuevo daño generado y que no se esté registrando cuando se agrega una vivienda o una familia, especificando la información general del daño y si pertenece a daños de infraestructura de viviendas o a daños de persona.	Visible
82	En el momento que se requiera el sistema debe ofrecer una opción que permita consultar los daños registrados.	Visible
83	El sistema debe ofrecer al usuario una opción que le permita modificar la información de un daño.	Visible
84	El sistema debe permitir al usuario eliminar un daño.	Visible

Funciones – Aplicación móvil

Req #	REQUERIMIENTOS FUNCIONALES	CAT
Usuarios		
85	El sistema deberá ofrecer al usuario una opción que le permita autenticarse e ingresar a la aplicación.	Visible
86	El sistema en el momento en que un usuario se autentique, deberá validar si el usuario existe o no en el sistema.	Oculto
Desastres		
87	El sistema debe permitir al usuario, registrar la información general del desastre generado: zona afectada y tipo de evento, fecha y hora de ocurrencia e impacto.	Visible
88	El sistema debe permitir consultar un desastre de acuerdo a la fecha o al código.	Visible
89	En el momento en que el usuario lo requiera el sistema debe ofrecerle una opción que le permita actualizar la información de un desastre.	Visible
90	En el momento que un usuario consulte un desastre, el sistema debe ofrecerle una opción que le permita registrar viviendas afectadas por el evento, especificando su información general como la dirección, el teléfono, el estrato, el formato de tenencia de la vivienda y si es rural o urbana; el estado de sus servicios	Visible

	públicos y daños ocasionados en su infraestructura.	
91	El sistema debe permitir al usuario en el momento en que registre una vivienda especificar las necesidades que esta posee, especificando cada tipo de necesidad y la cantidad de recursos que requiere para suplirla.	Visible
92	El sistema debe permitir al usuario registrar la evaluación agropecuaria y de cultivos de una vivienda, especificando el tipo de cultivos y las pérdidas económicas y materiales.	Visible
93	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las viviendas afectadas por un desastre, visualizando por cada una su información básica.	Visible
94	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las necesidades de una vivienda.	Visible
95	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar la evaluación agropecuaria de una vivienda.	Visible
96	El sistema debe permitir al usuario actualizar la información de una vivienda.	Visible
97	El sistema debe permitir al usuario actualizar la información de las necesidades de una vivienda.	Visible
98	El sistema debe permitir al usuario actualizar la información de la evaluación agropecuaria de una vivienda.	Visible
99	En el momento en un usuario registre una vivienda, el sistema debe ofrecerle una opción que le permita registrar la familia que habita en ella, especificando su información general como sus apellidos, jefe de familia y tipo de comunidad.	Visible
100	El sistema debe permitirle al usuario, una vez registre una familia, especificar las necesidades que ella posee, indicando que necesidades son y la cantidad de recursos que requiere para suplirlas.	Visible
101	En el momento en que se haya registrado una familia, el sistema debe ofrecerle al usuario una opción que le permita registrar las personas que pertenecen a ella, especificando por cada una: su identificación, nombre, apellido, edad, sexo, ocupación, el parentesco con el jefe de la familia, además de su estado de salud y los daños o enfermedades que le haya ocasionado el desastre.	Visible
102	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las familias que pertenecen a una vivienda, visualizando por cada una su información básica.	Visible
103	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las	Visible

	necesidades de una familia.	
104	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las personas que pertenecen a una familia.	Visible
105	El sistema debe permitir al usuario actualizar la información de una familia.	Visible
106	El sistema debe permitir al usuario actualizar la información de las necesidades de una familia.	Visible
107	El sistema debe permitir al usuario actualizar la información de las personas que pertenecen a una familia.	Visible

Funciones – Aplicación móvil de respaldo


Req #	REQUERIMIENTOS FUNCIONALES	CAT
Desastres		
108	El sistema debe permitir al usuario, registrar la información general del desastre generado: zona afectada y tipo de evento, fecha y hora de ocurrencia e impacto.	Visible
109	El sistema debe permitir consultar un desastre de acuerdo a la fecha o al código.	Visible
110	En el momento en un usuario consulte un desastre, el sistema debe ofrecerle una opción que le permita registrar viviendas afectadas por el evento, especificando su información general como la dirección, el teléfono, el estrato, el formato de tenencia de la vivienda y si es rural o urbana; el estado de sus servicios públicos y daños ocasionados en su infraestructura.	Visible
111	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las viviendas afectadas por un desastre, visualizando por cada una su información básica.	Visible
112	En el momento en un usuario registre una vivienda, el sistema debe ofrecerle una opción que le permita registrar la familia que habita en ella, especificando su información general como sus apellidos, jefe de familia y tipo de comunidad.	Visible
113	En el momento en que se haya registrado una familia, el sistema debe ofrecerle al usuario una opción que le permita registrar las personas que pertenecen a ella, especificando por cada una: su identificación, nombre, apellido, edad, sexo, ocupación, el parentesco con el jefe de la familia, además de su estado de salud y los daños o enfermedades que le haya ocasionado el desastre.	Visible
114	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las familias que pertenecen a una vivienda, visualizando por	Visible

	cada una su información básica.	
115	En el momento en que un usuario lo requiera el sistema deberá ofrecerle una opción que le permita consultar las personas que pertenecen a una familia.	Visible
116	En el momento en que un usuario lo requiera y exista conexión a la aplicación central, el sistema debe ofrecer una opción que le permita enviar los datos registrados en el móvil al servidor central.	Visible
117	En el momento en que un usuario vaya a enviar información desde el dispositivo a la aplicación central, el usuario debe especificar su nombre y login para poder enviar la información.	Visible
118	En el momento en que un usuario envíe información al servidor desde el móvil éste debe validar que el usuario es correcto antes de registrar los datos.	Oculto

Funciones – Aplicación Espacial

Req #	REQUERIMIENTOS FUNCIONALES	CAT
119	La aplicación deberá permitir al usuario la visualización de un mapa, parcial o total, de la superficie de la ciudad.	Visible
120	El sistema deberá ofrecer una opción que le permita al usuario visualizar u ocultar las diferentes capas de información de división territorial, las cuales son: direcciones, barrios, zonas verdes, manzanas, ríos y comunas.	Visible
121	La aplicación deberá permitir al usuario visualizar en el mapa los desastres de acuerdo a los siguientes parámetros: tipo, ocurrencia en un periodo de tiempo determinado ó área de la localidad (barrio).	Visible
122	En el momento en que el usuario lo requiera el sistema deberá permitirle realizar las acciones acercar, alejar y desplazarse en el mapa.	Visible
123	El sistema debe ofrecer al usuario la opción de reiniciar el mapa.	Visible
124	La aplicación deberá permitirle al usuario visualizar la información general de los desastres ubicados en el mapa, cuando éste pasa el mouse por ellos.	Visible

6.1.2 Requerimientos No Funcionales

	Especificación de Requerimientos ISO-9000 Universidad del Valle	Documento: ER-001	Revisión: 00 Fecha: 12/09/07
Título: Especificación de Requerimientos no funcionales para el desarrollo del sistema de información web para el manejo de información en el CLOPAD - SIWEPAD			

Requerimientos – Aplicación General y móvil

ATRIBUTOS	ESPECIFICACIÓN DE HARDWARE Y SOFTWARE
Plataforma de Software	Servidor <ul style="list-style-type: none"> • Sistema Operativo: Windows Service Pack 2 • Servidor Web Tomcat • SMBD: PostgreSQL 8.2 • jdk-6-windows-i586 Cliente de escritorio <ul style="list-style-type: none"> • Navegador Web: IE 6.0 Service Pack 2, Firefox 2.0, Recomendado IE • Interprete en el navegador de scripts (JavaScripts) Cliente móvil <ul style="list-style-type: none"> • Navegador Internet Explorer Mobile o Firefox Mobile
Plataforma de Hardware	Equipo de Cómputo: Procesador: AMD Athlon XP, 1.8 GHz Disco Duro: 80 GB Memoria RAM: 1 GB Tarjeta de Red: Fast Ethernet VIA PCI 10/100Mb

Requerimientos – Aplicación móvil de respaldo

ATRIBUTOS	ESPECIFICACIÓN DE HARDWARE Y SOFTWARE
Plataforma de Software	Dispositivo Workabout Pro C, M - E Sistema Operativo: Windows Mobile 2003 SE
Plataforma de Hardware	Equipo de Cómputo: Memory : 64MB Flash, 64MB RAM Processor : Intel Xscale PXA255 400MHz Ingress Protection Rating : IP54 Expansion : <ul style="list-style-type: none"> - 1 x SD/MMC slot - 100 pin high-speed expansion connector - Type II CompactFlash slot

Requerimientos – Aplicación Espacial

ATRIBUTOS	ESPECIFICACIÓN DE HARDWARE Y SOFTWARE
Plataforma de Software	<p>Servidor</p> <ul style="list-style-type: none"> • Sistema Operativo: Windows Service Pack 2 • Servidor de Mapas Mapserver - ms4w-2.2.7 • Servidor de Base de Datos Espacial: PostgreSQL 8.0 y PostGIS <p>Cliente</p> <ul style="list-style-type: none"> • Navegador Web: IE 6.0, Firefox 2.0, Recomendado IE • Interprete en el navegador de scripts (JavaScripts)
Plataforma de Hardware	<p>Equipo de Cómputo: Procesador: AMD Athlon XP, 1.8 GHz Disco Duro: 80 GB Memoria RAM: 1 GB Tarjeta de Red: Fast Ethernet VIA PCI 10/100Mb</p>

6.2 DEFINICIÓN DE CASOS DE USO

6.2.1 Diagramas de Casos de Uso

Diagrama de Caso de Uso – Ingresar al Sistema

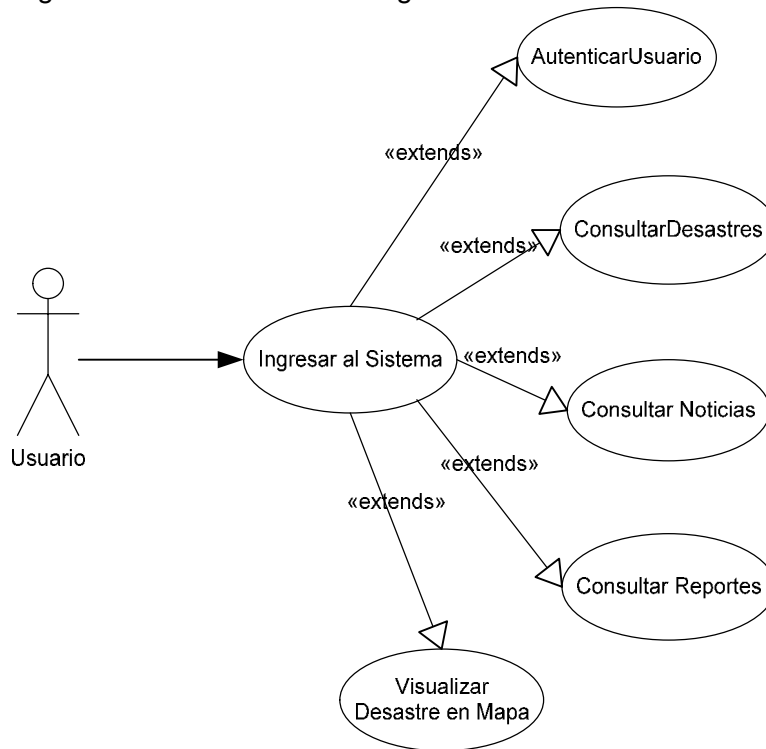


Diagrama de Casos de Uso – Gestión de Usuarios

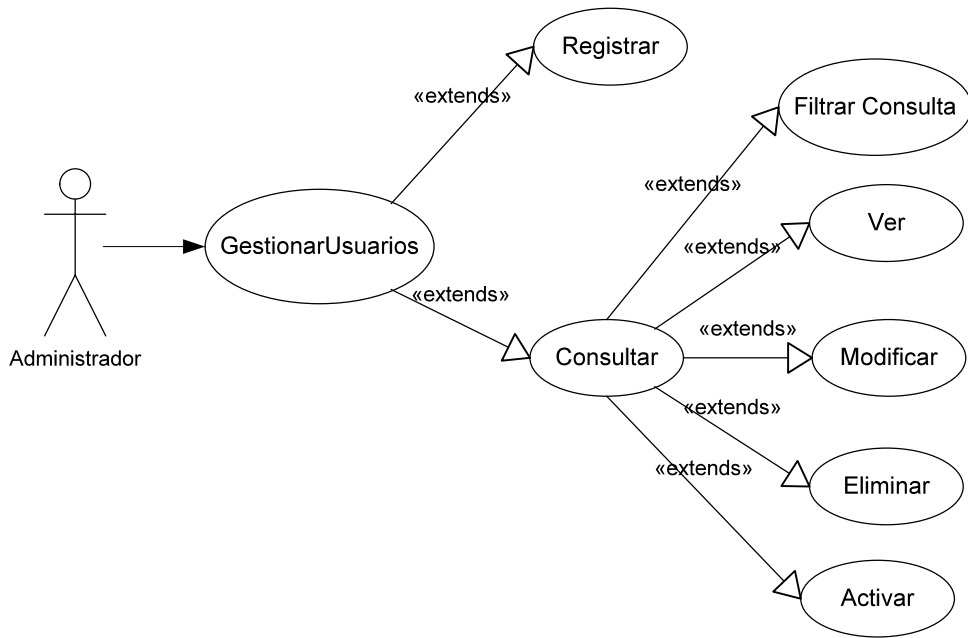


Diagrama de Casos de Uso – Gestión de Grupos

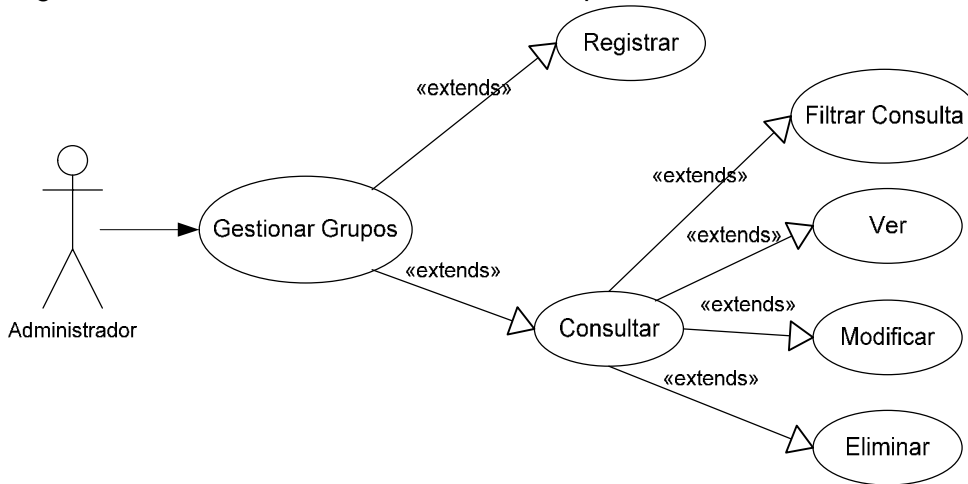


Diagrama de Casos de Uso – Gestión de Perfiles

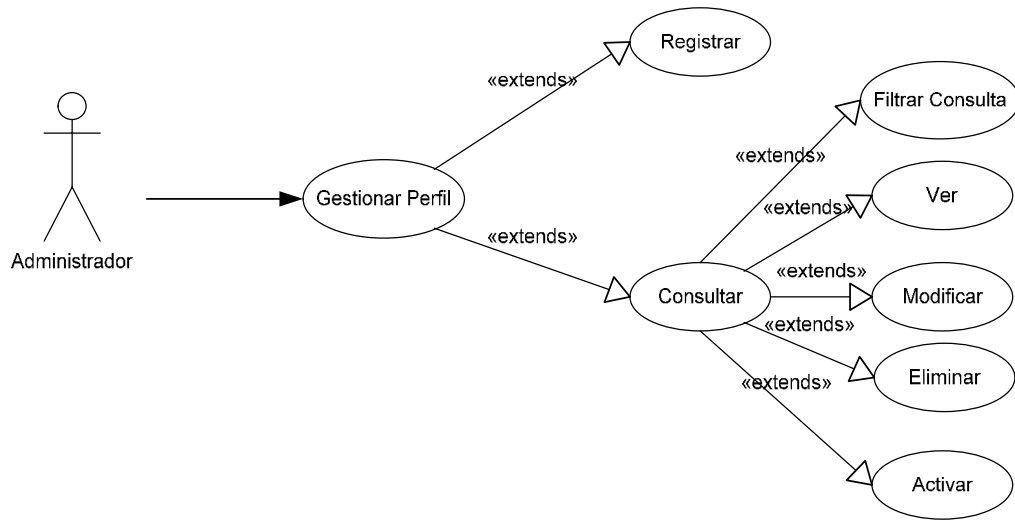


Diagrama de Casos de Uso – Módulo Recursos

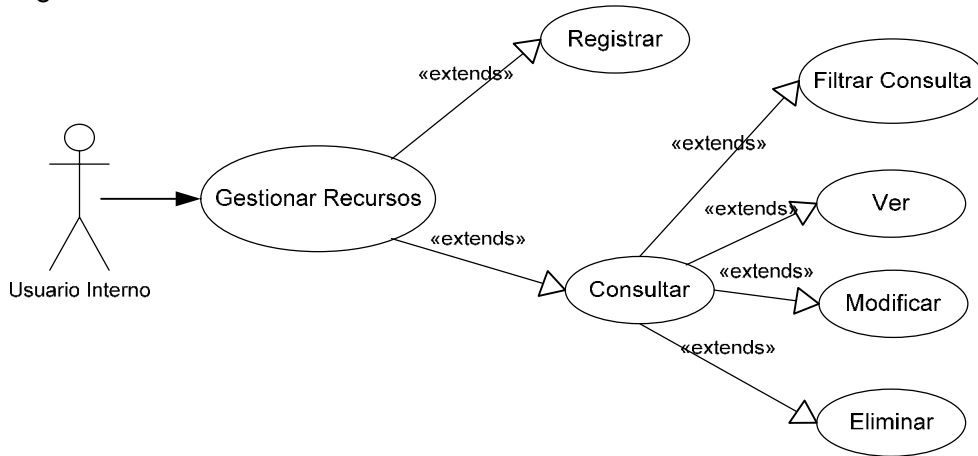
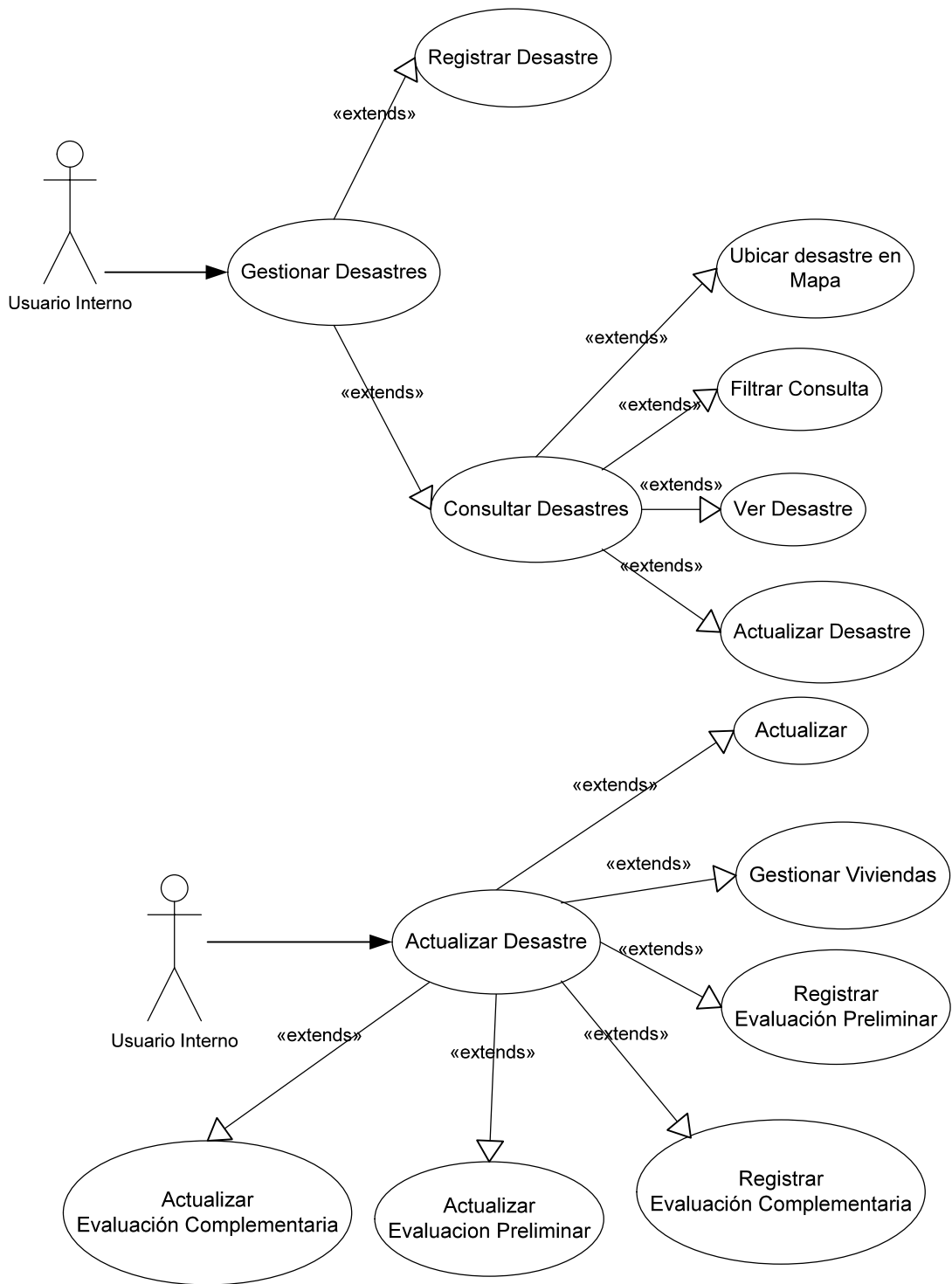
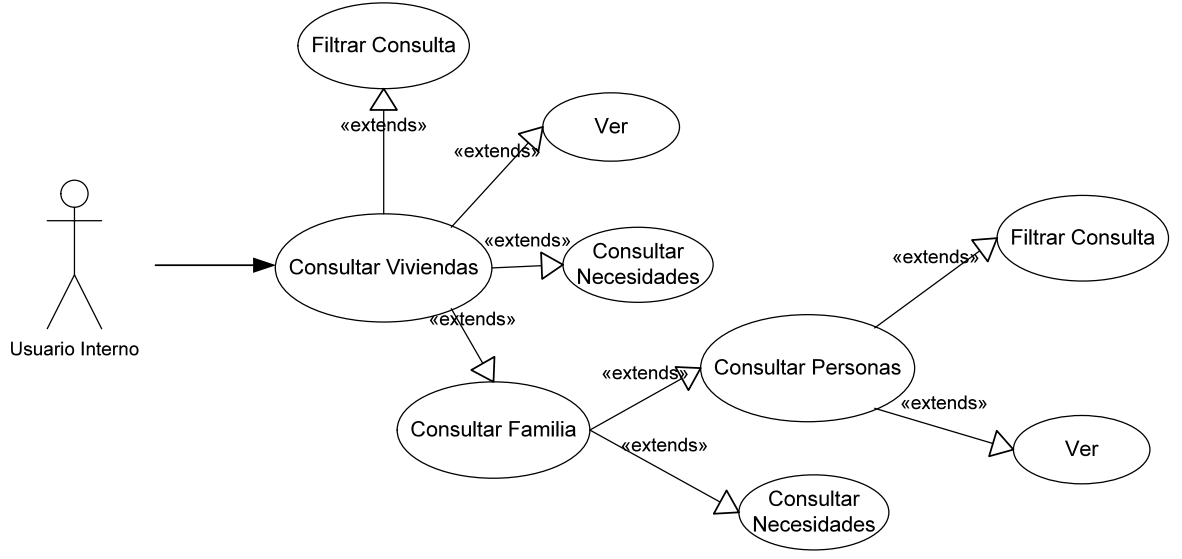
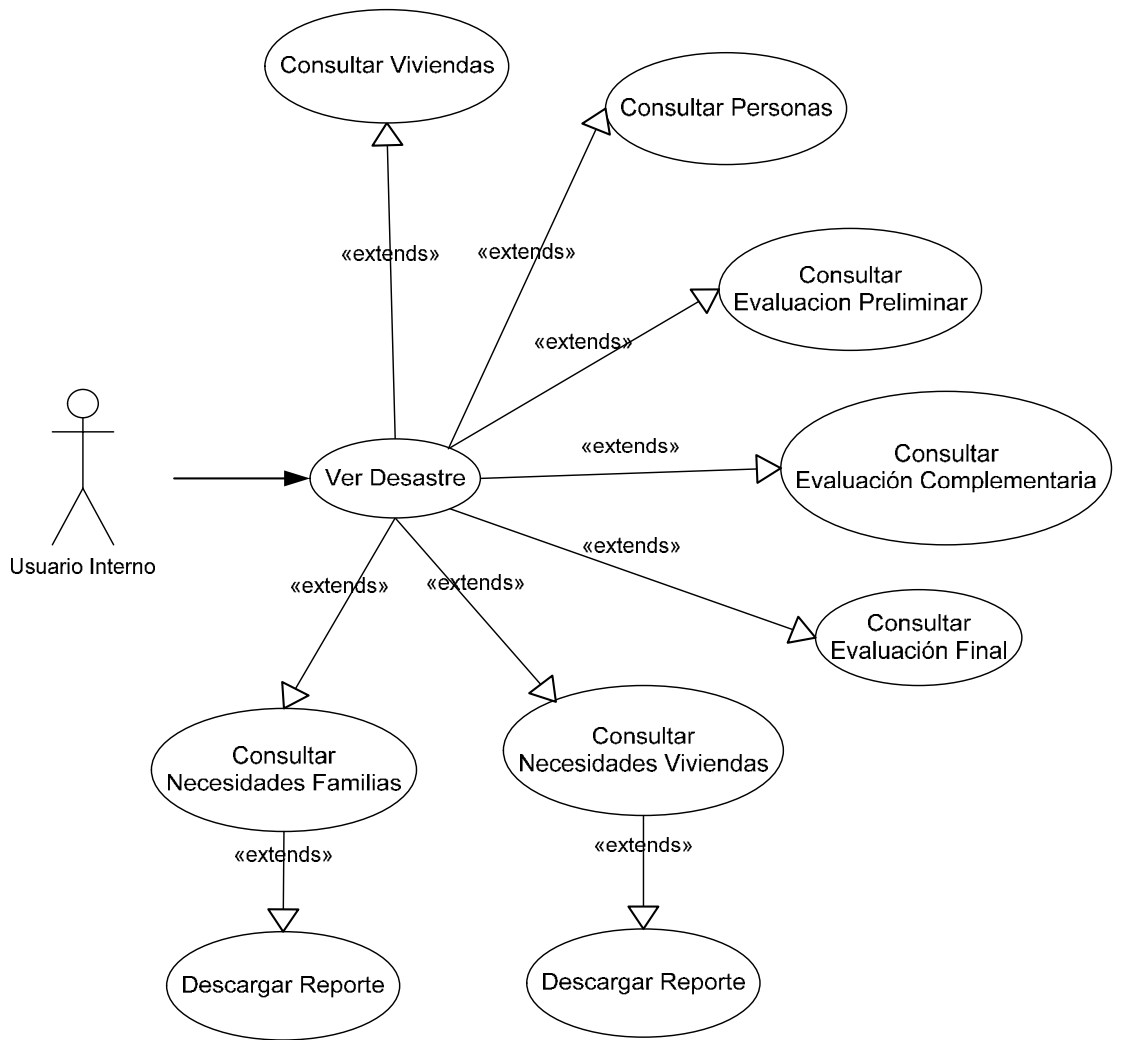


Diagrama de Casos de Uso – Módulo Desastres





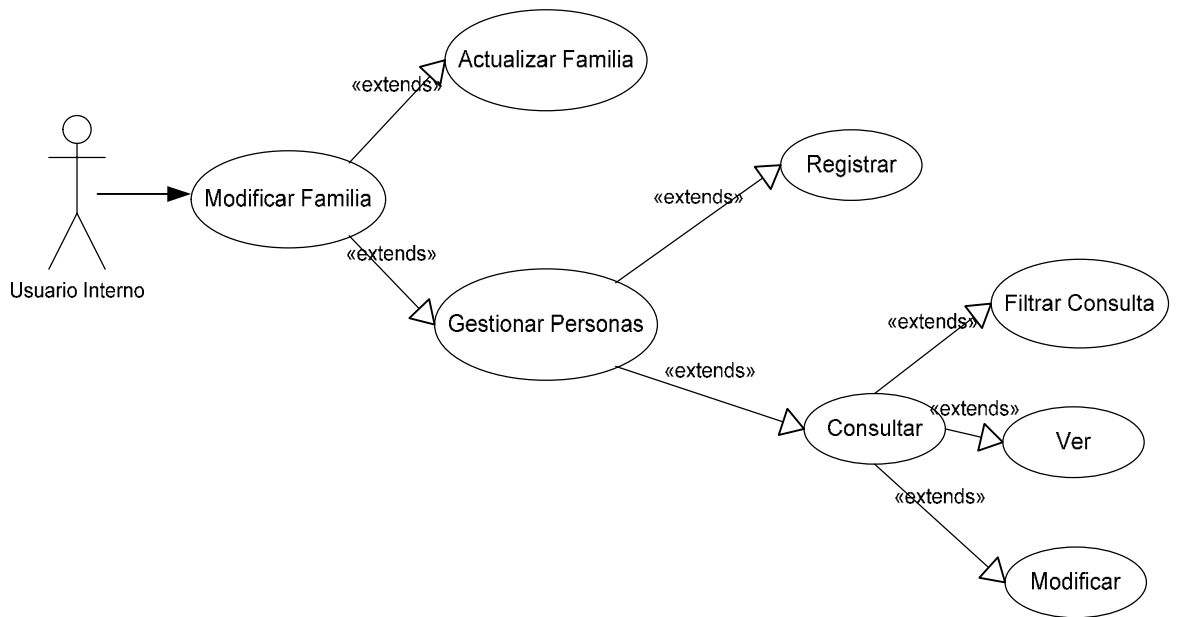
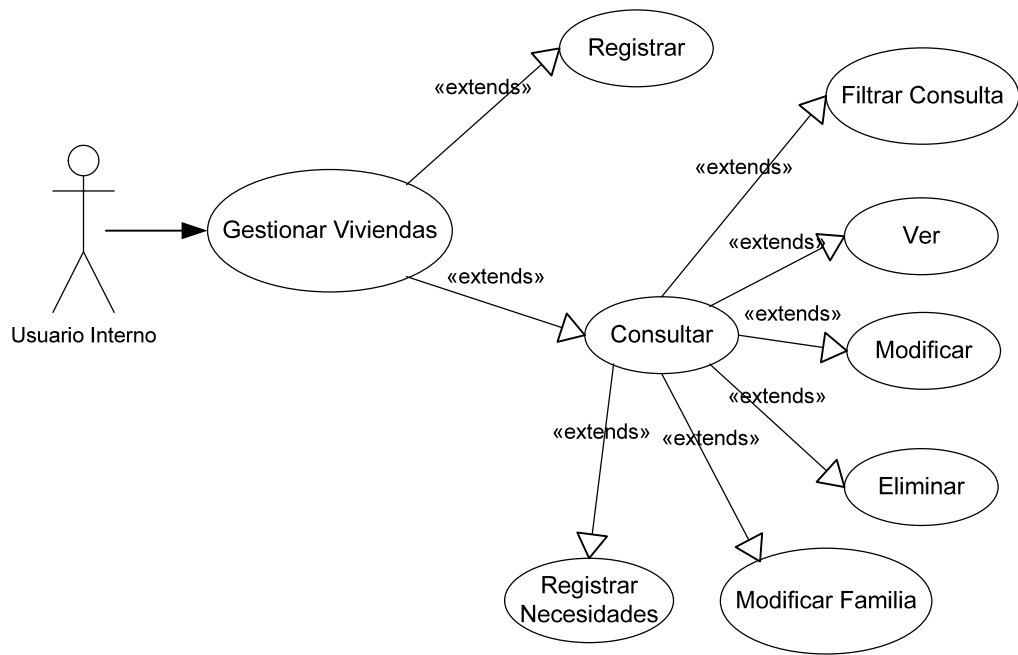


Diagrama de Casos de Uso – Módulo Reportes

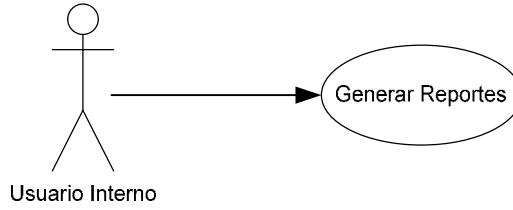
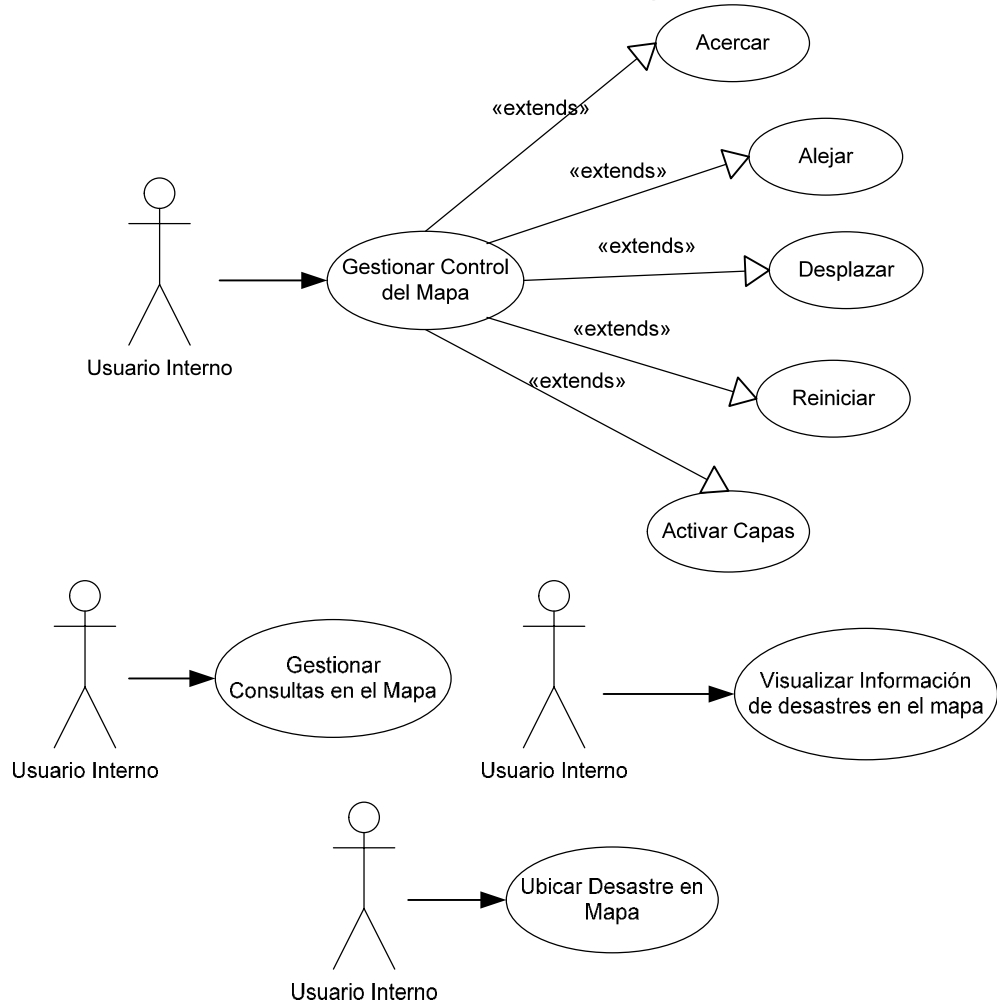
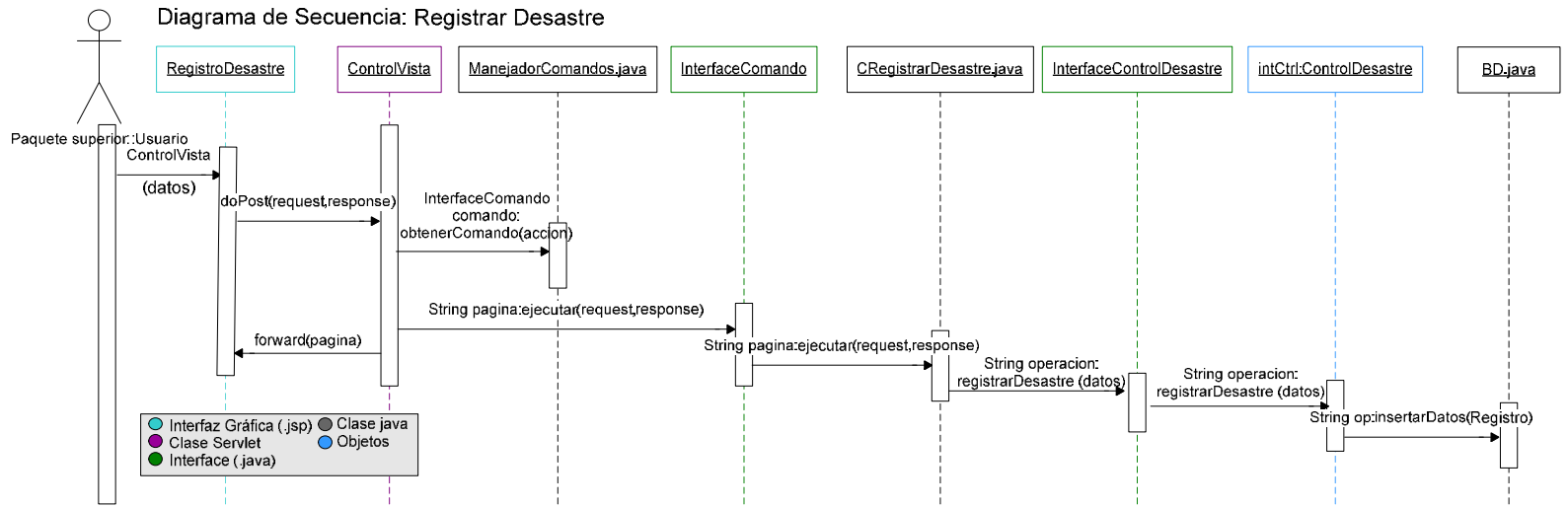


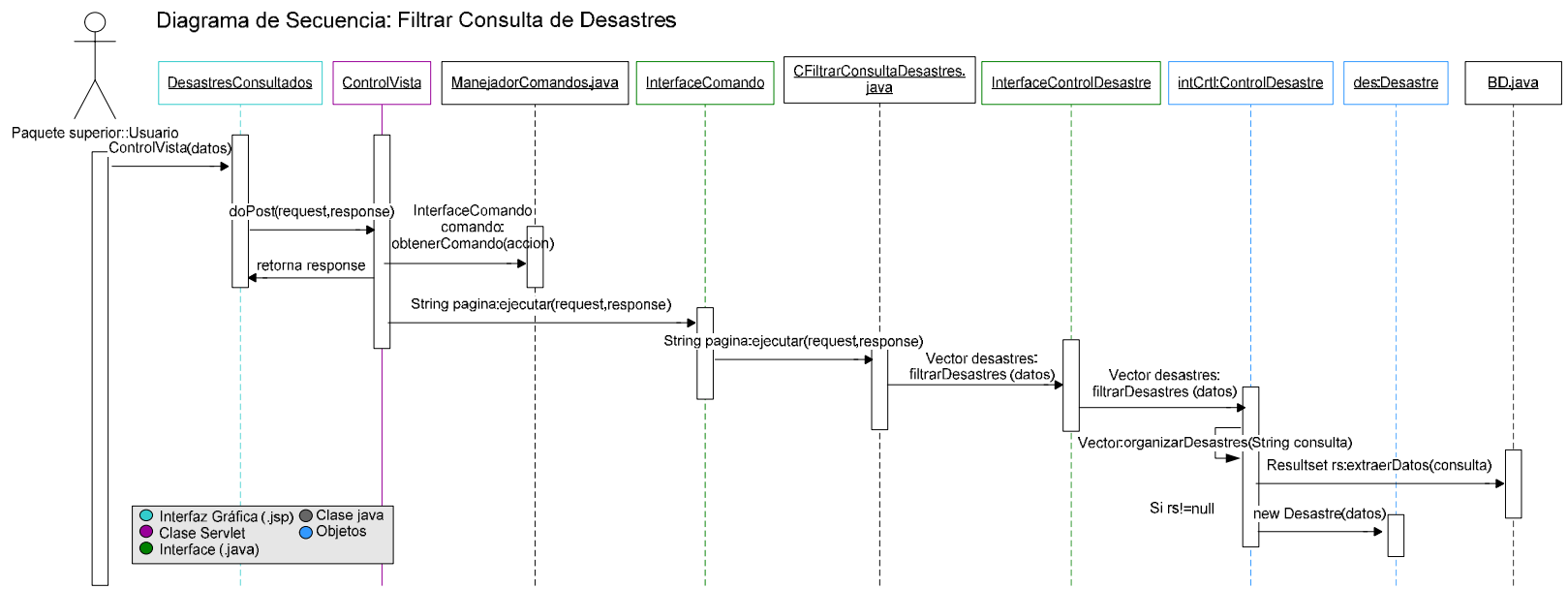
Diagrama de Casos de Uso – Módulo de Información Espacial

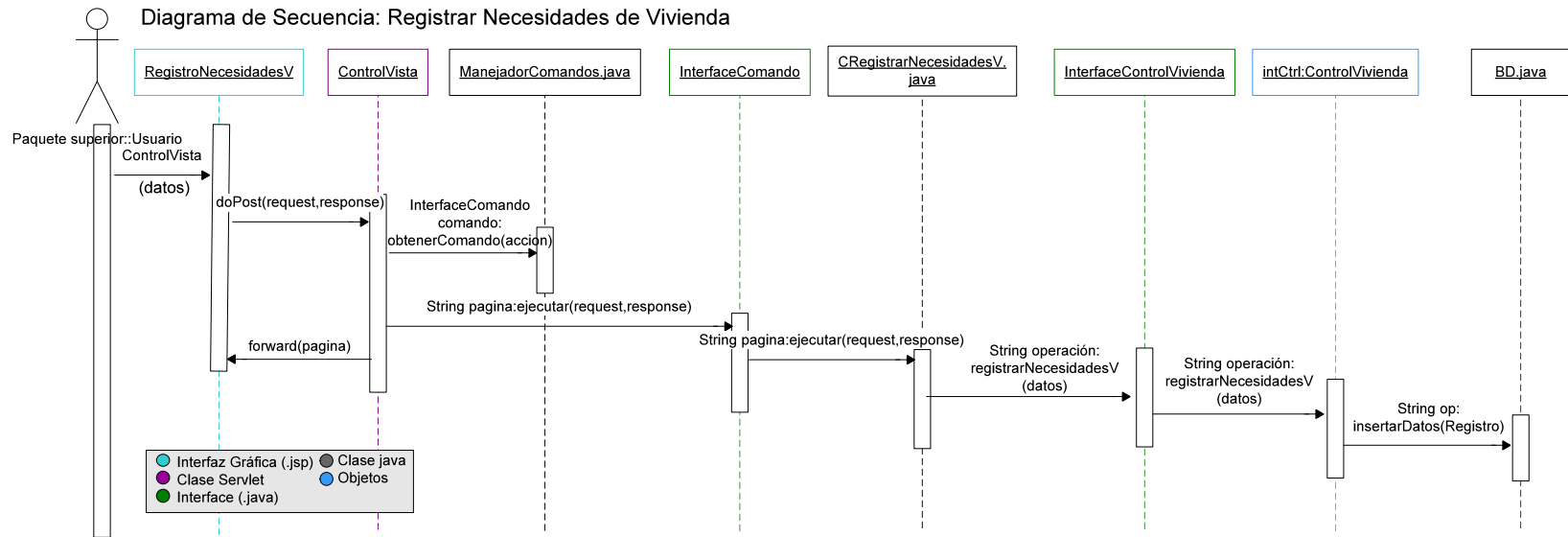
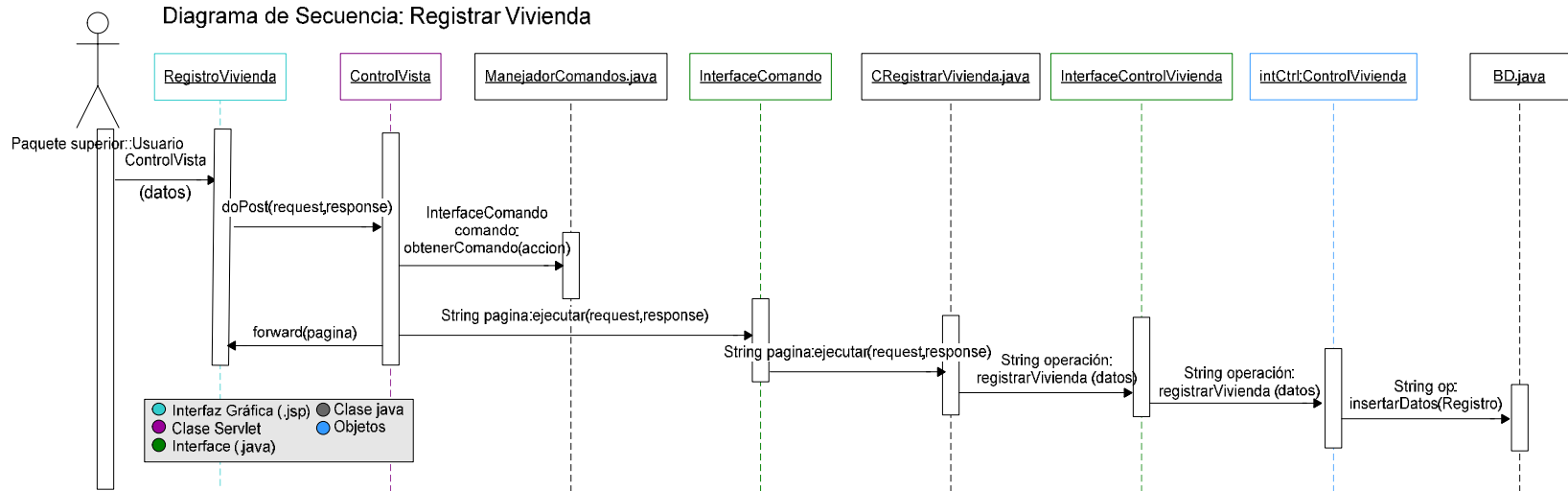


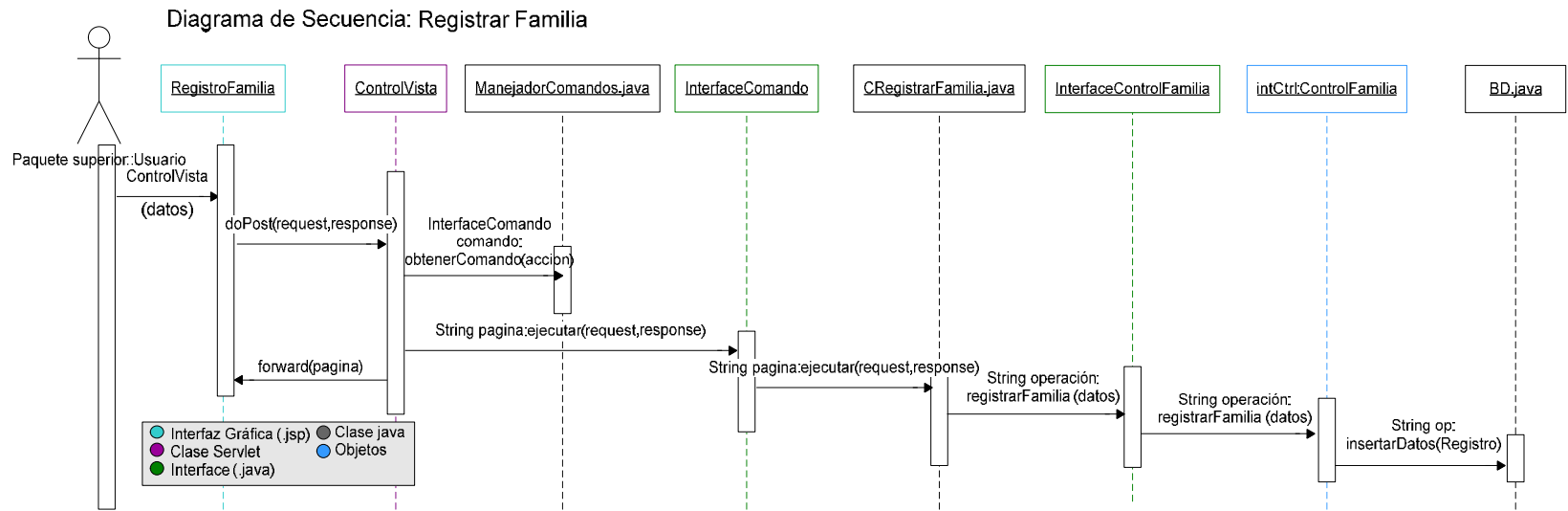
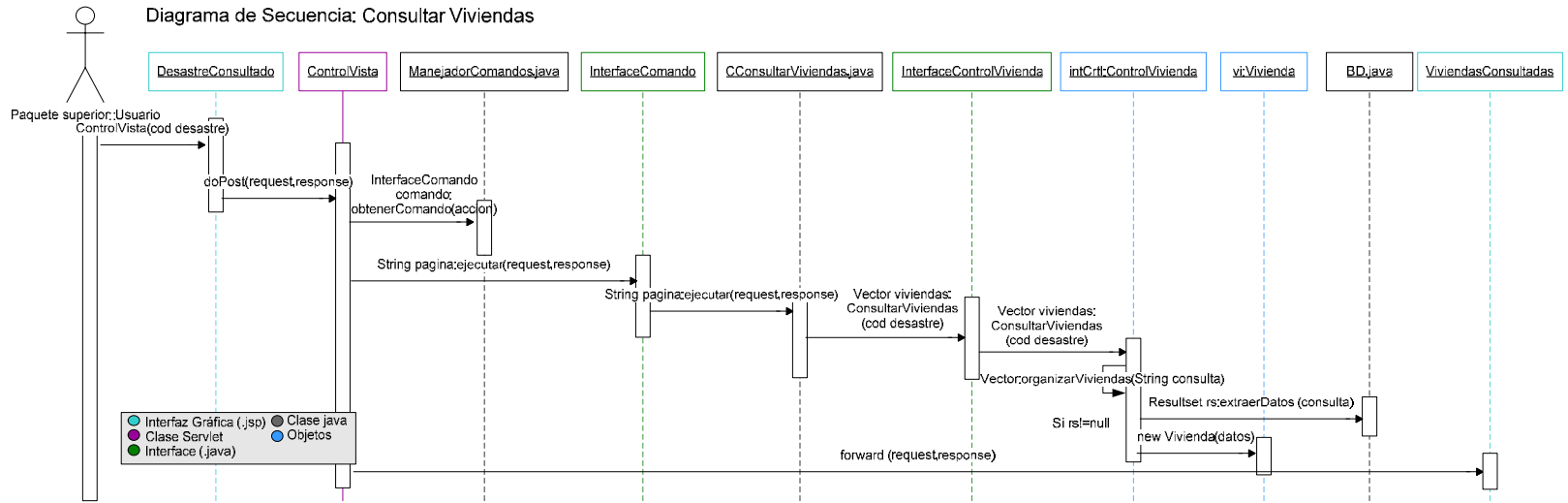
6.3 DIAGRAMAS DE SECUENCIA

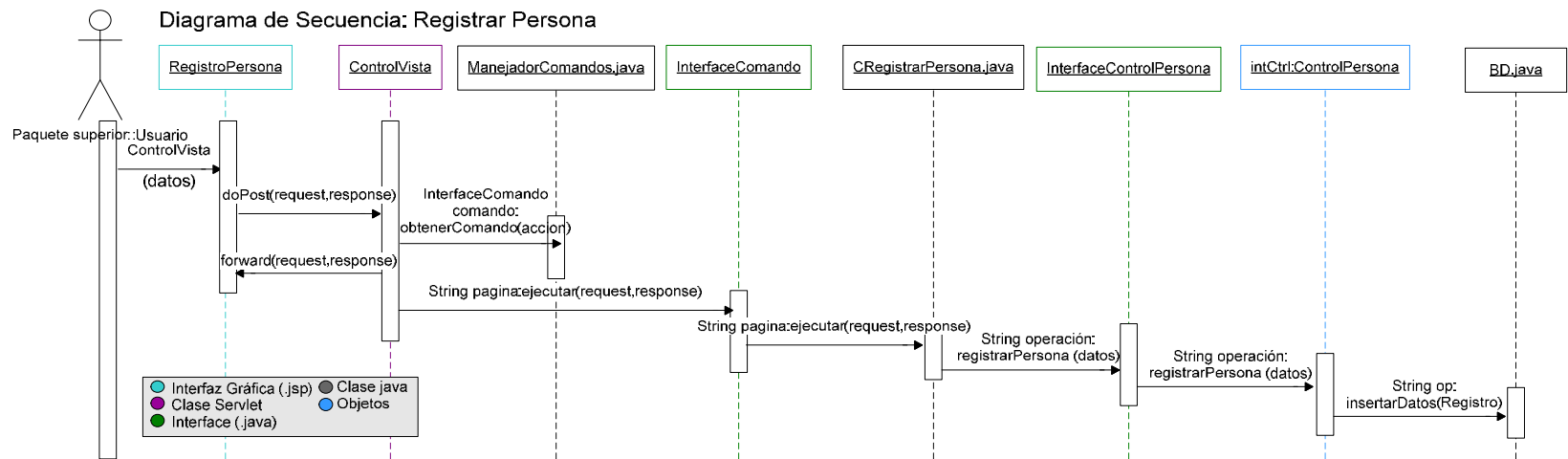
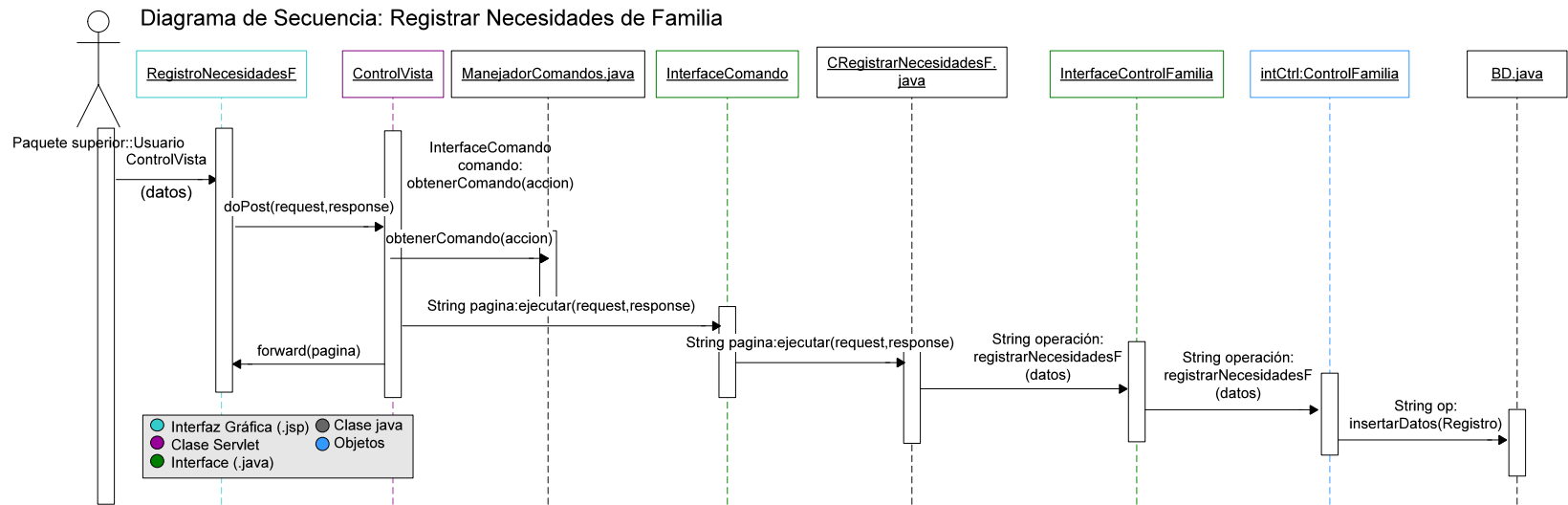
6.3.1 Aplicación web general

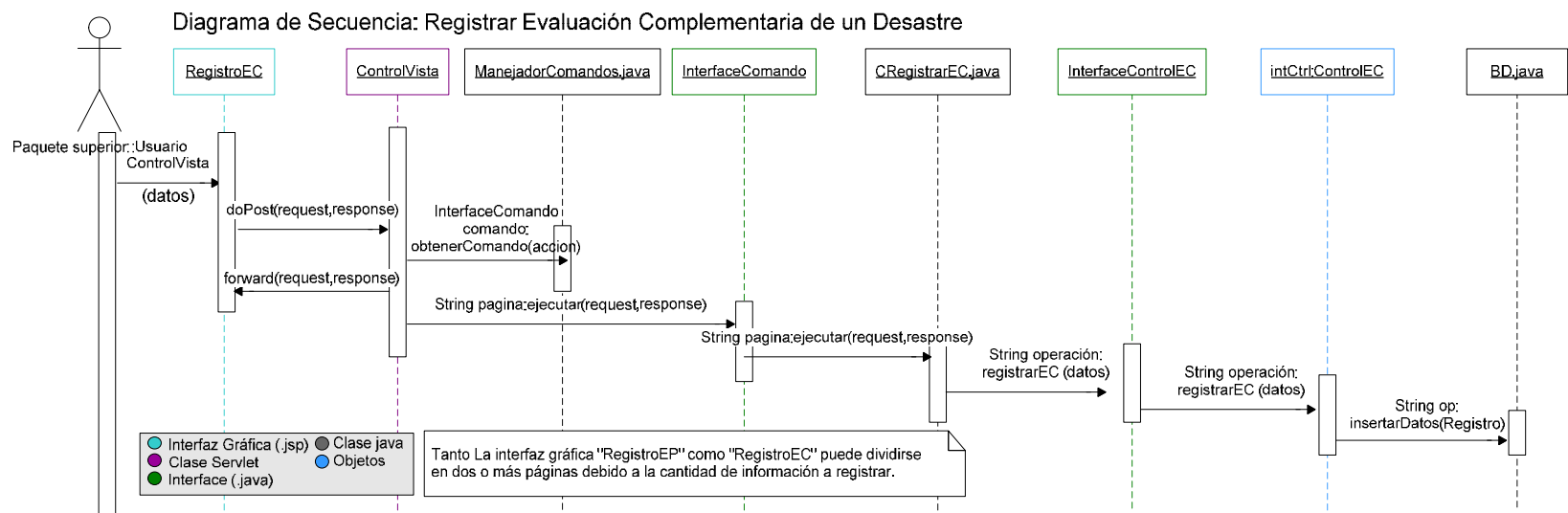
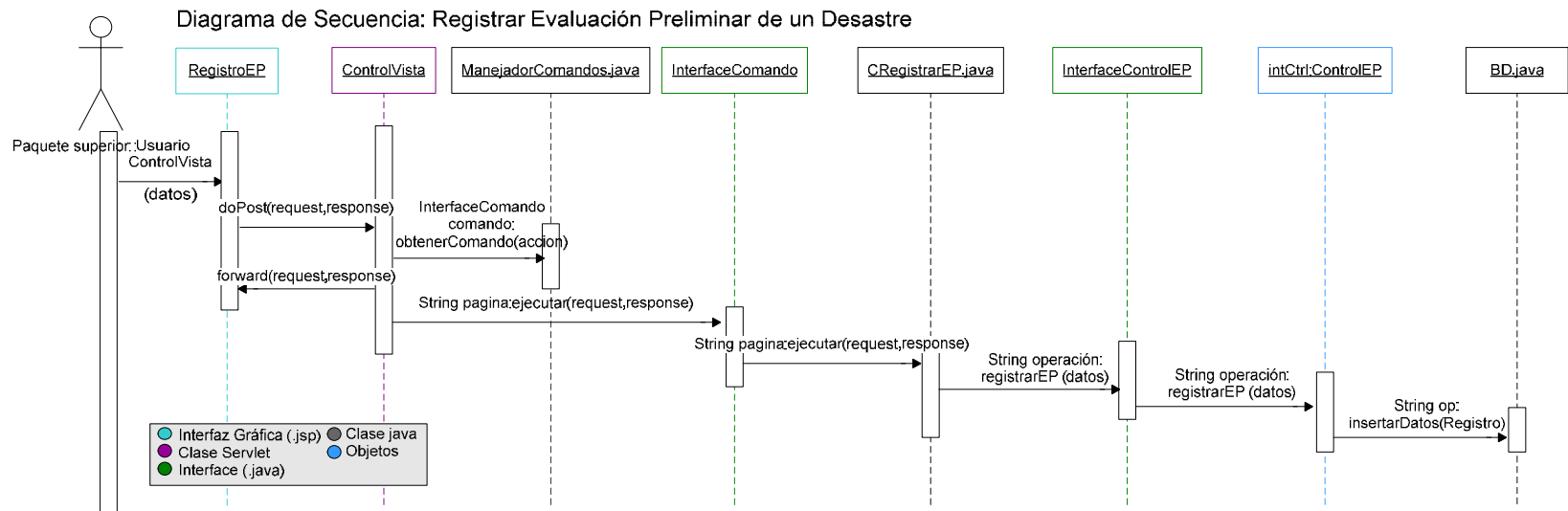


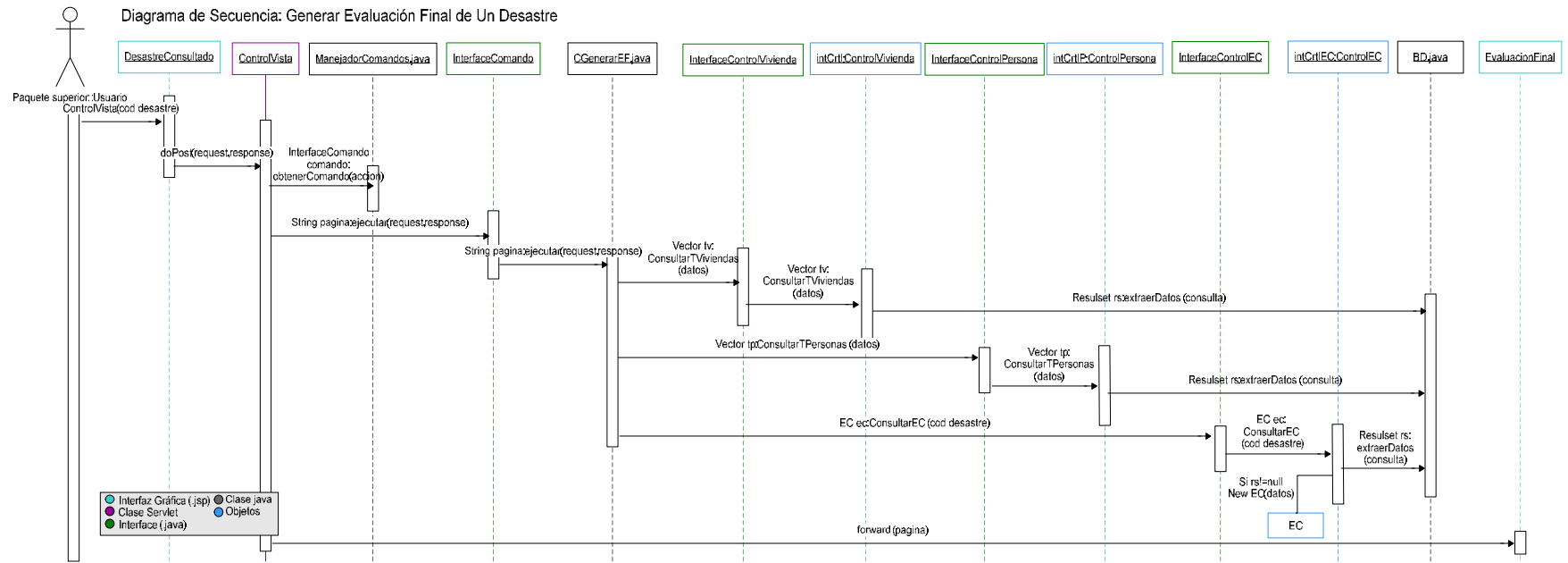












6.3.2 Aplicación de Respaldo (J2ME)

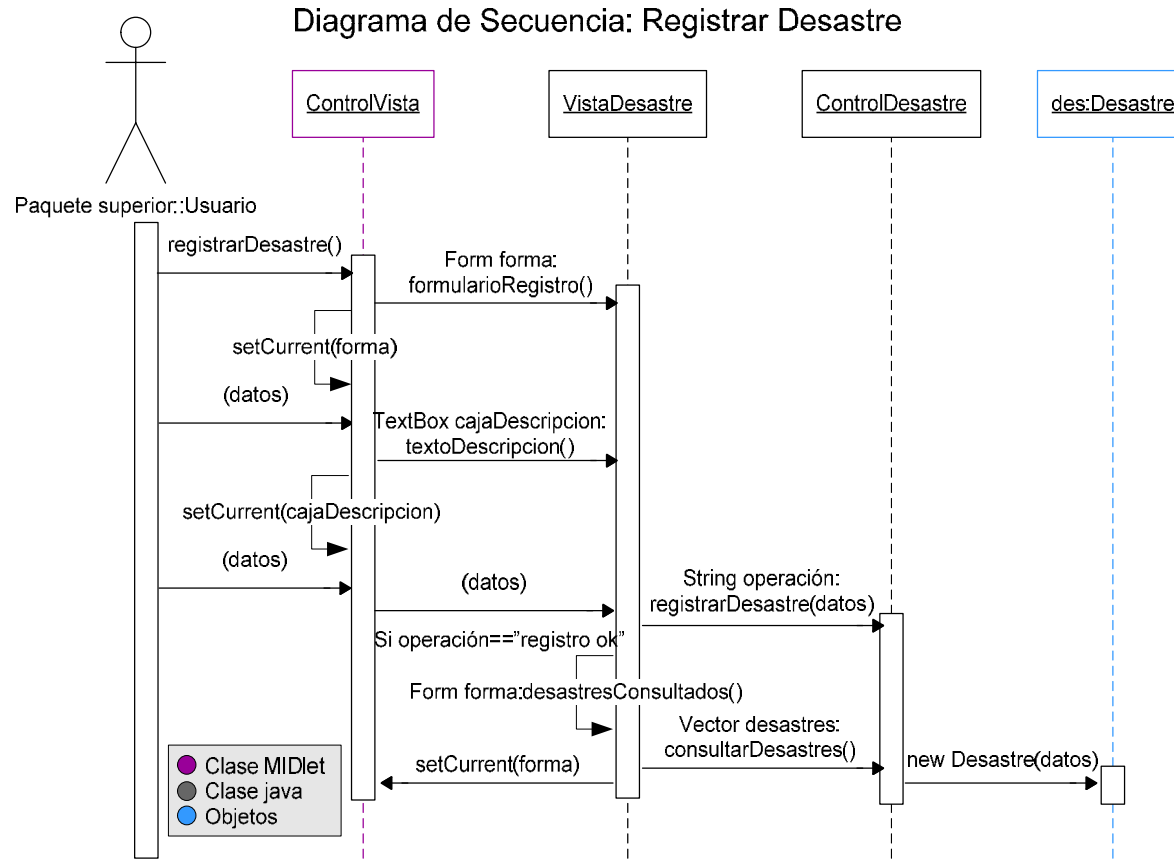


Diagrama de Secuencia: Registrar Vivienda

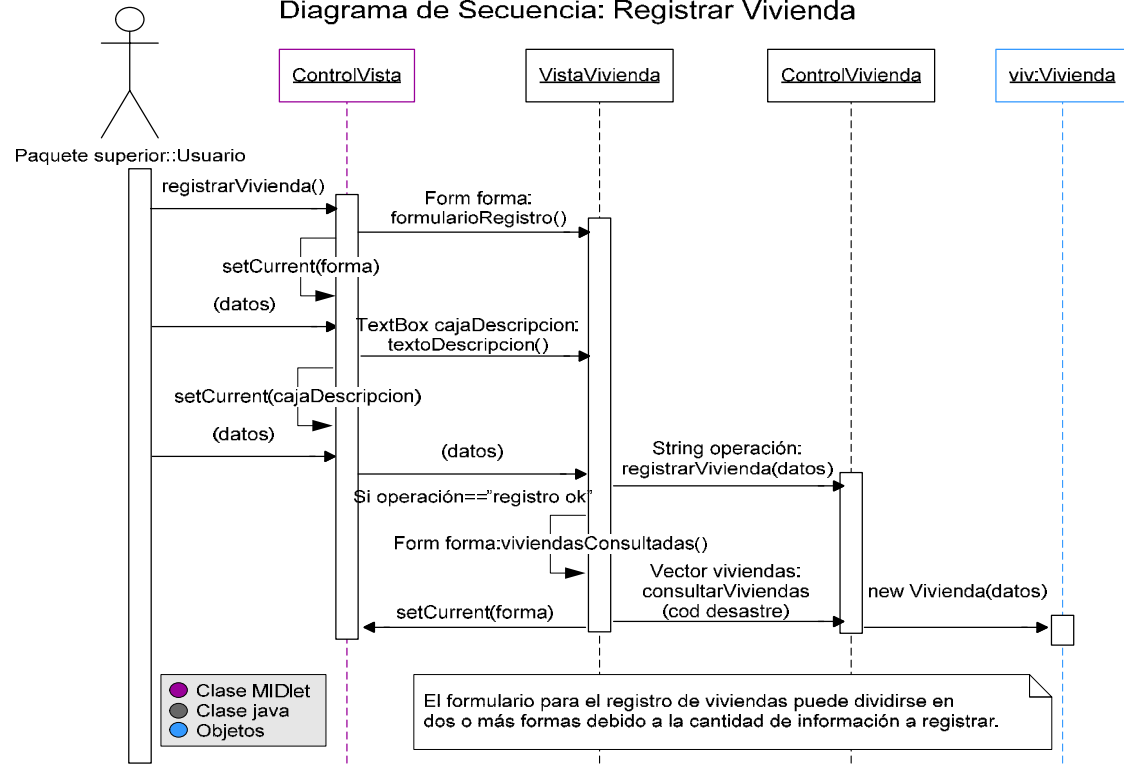


Diagrama de Secuencia: Registrar Familia

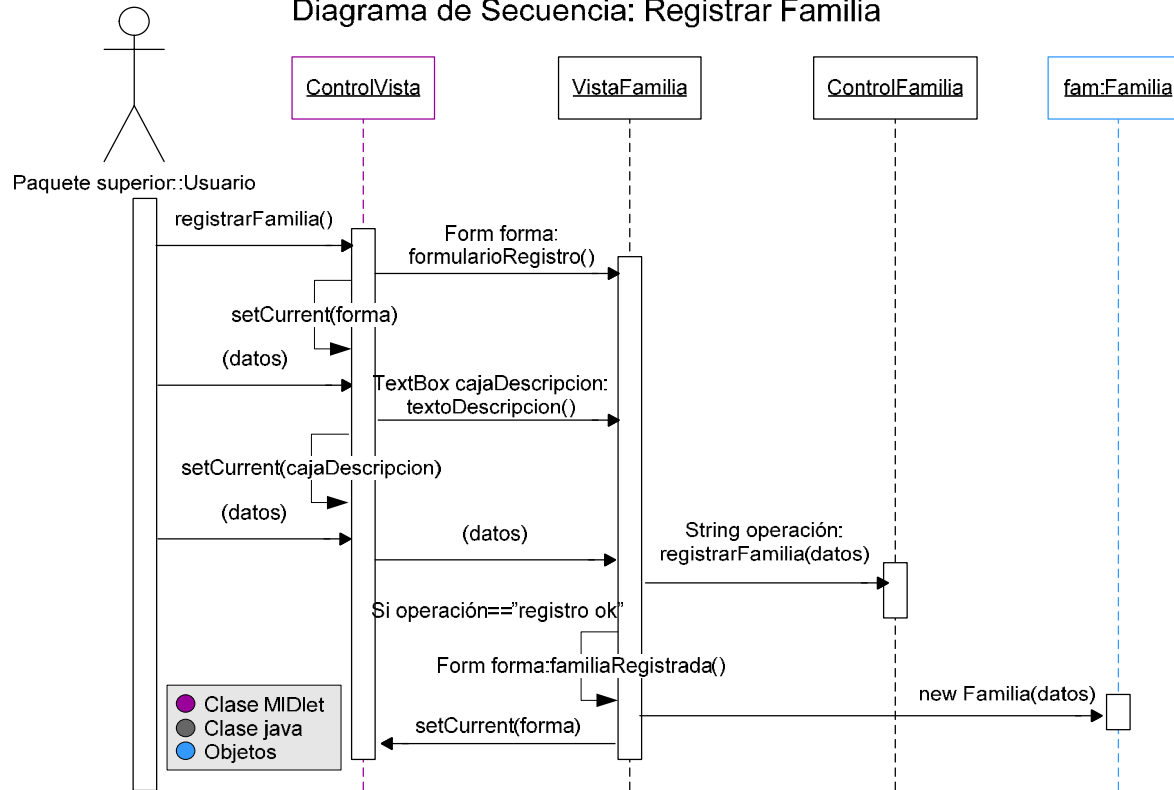
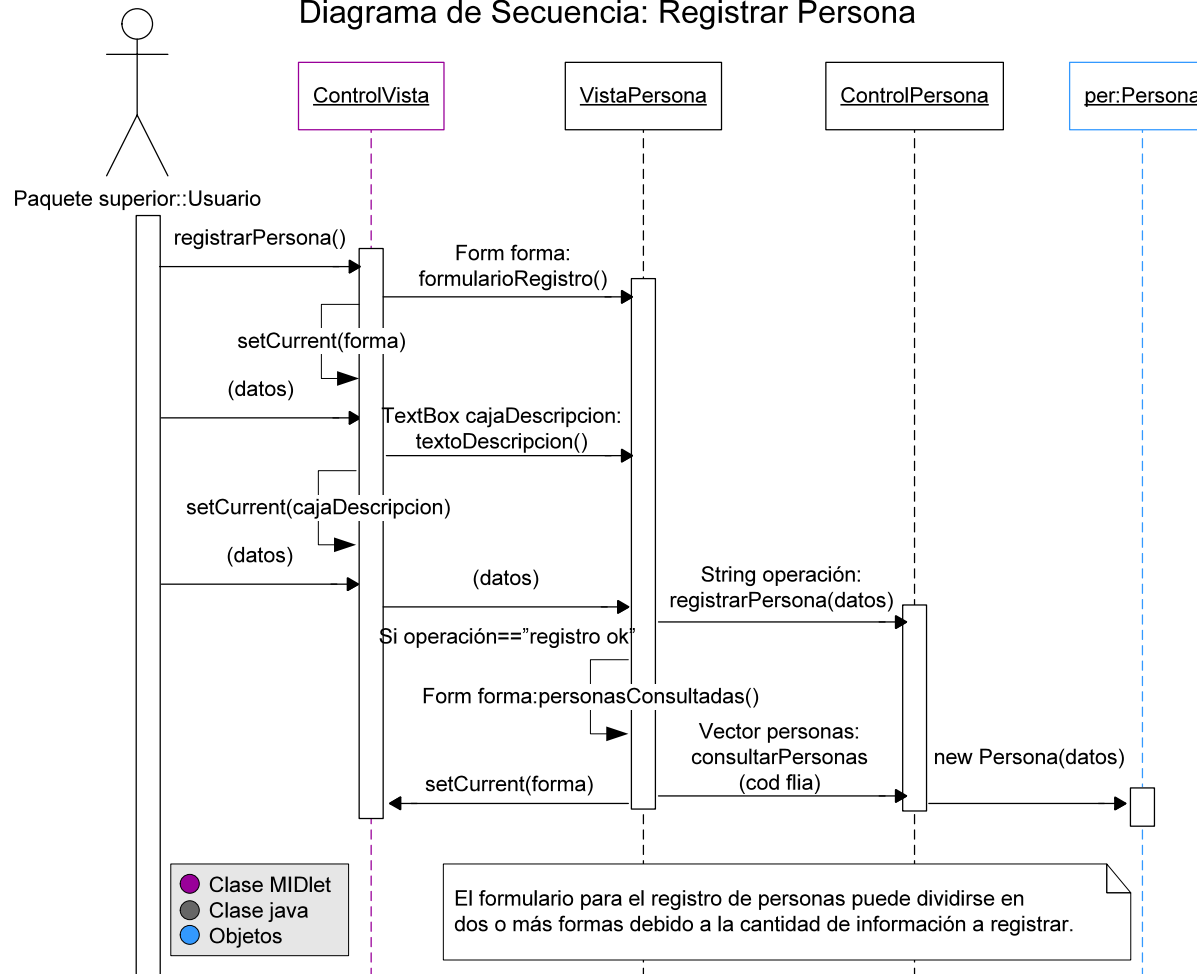


Diagrama de Secuencia: Registrar Persona



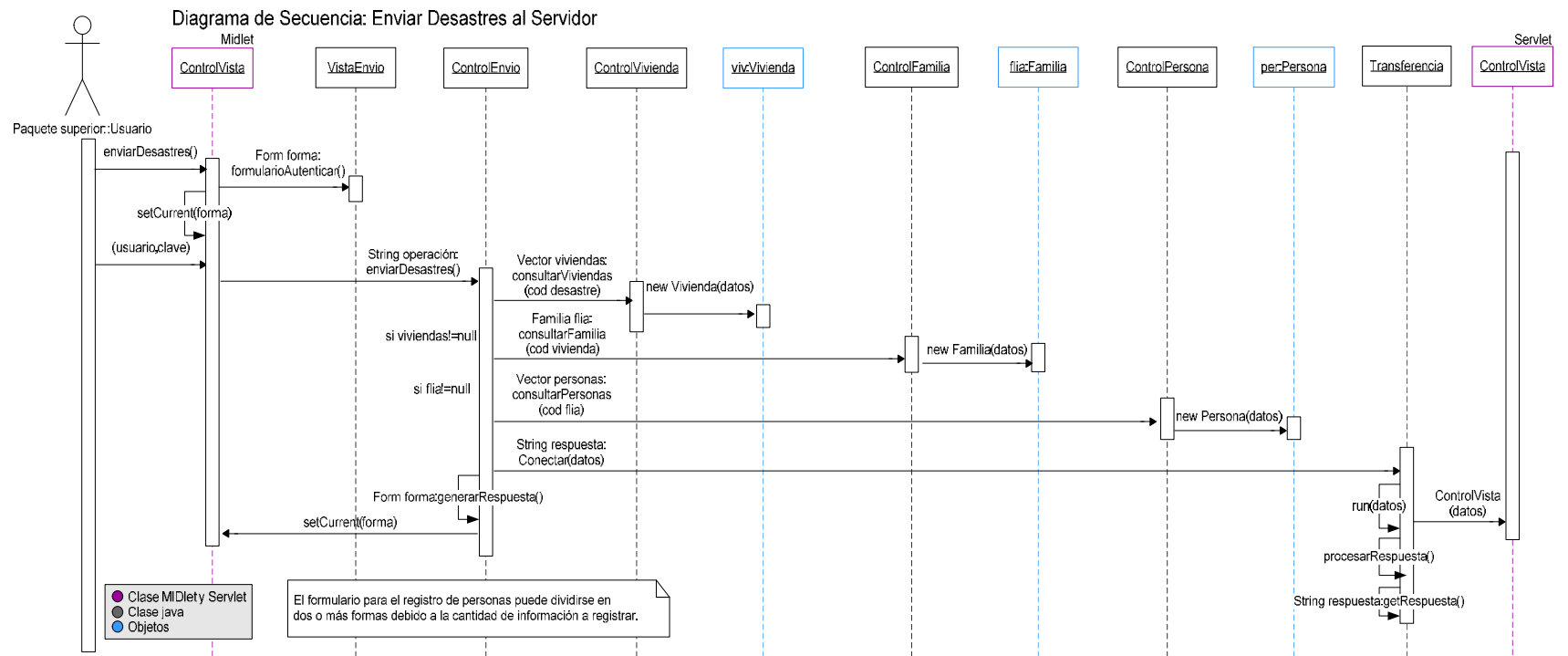
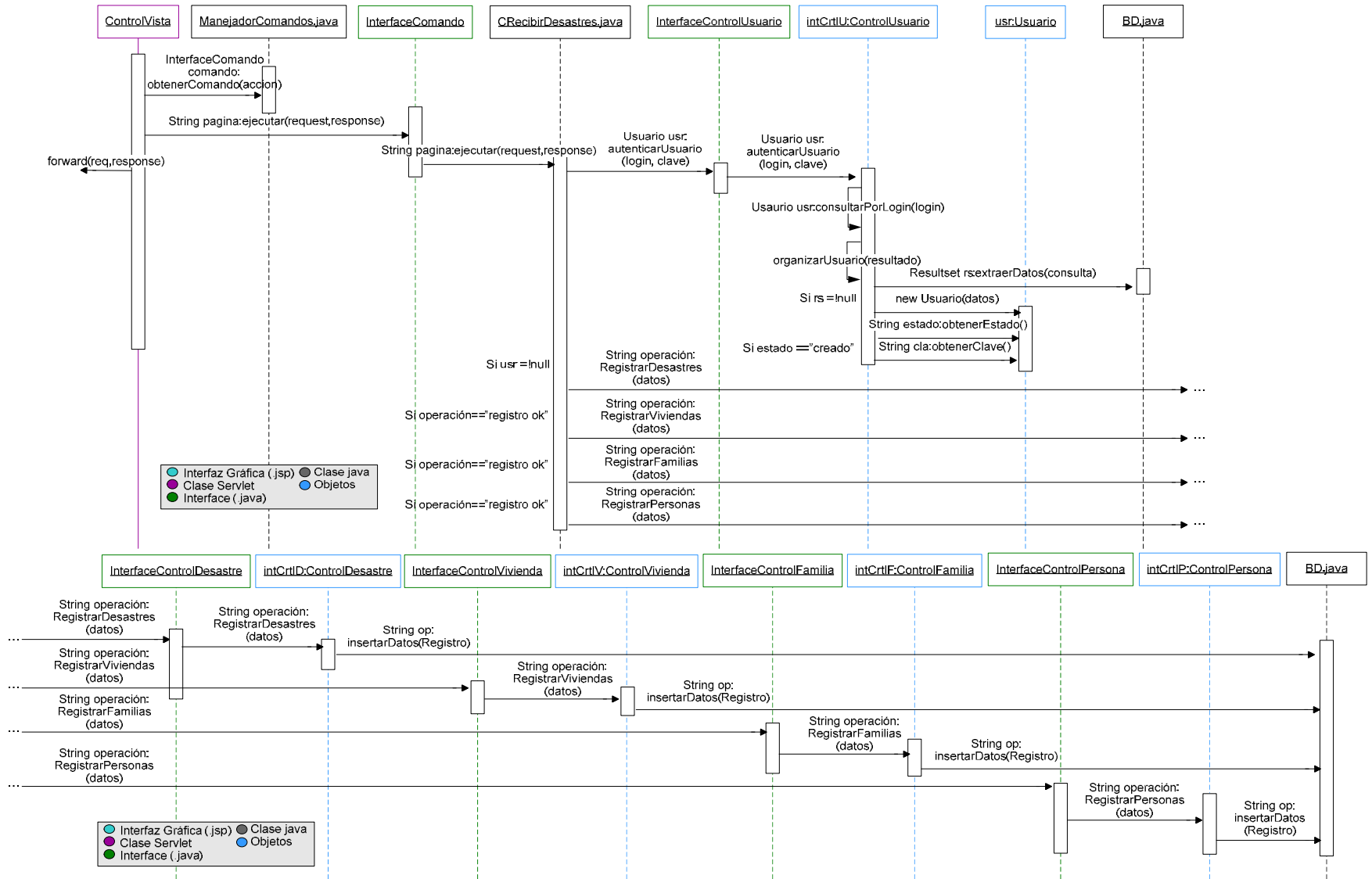


Diagrama de Secuencia: Recibir Desastres de dispositivo móvil



6.3.3 Aplicación Espacial

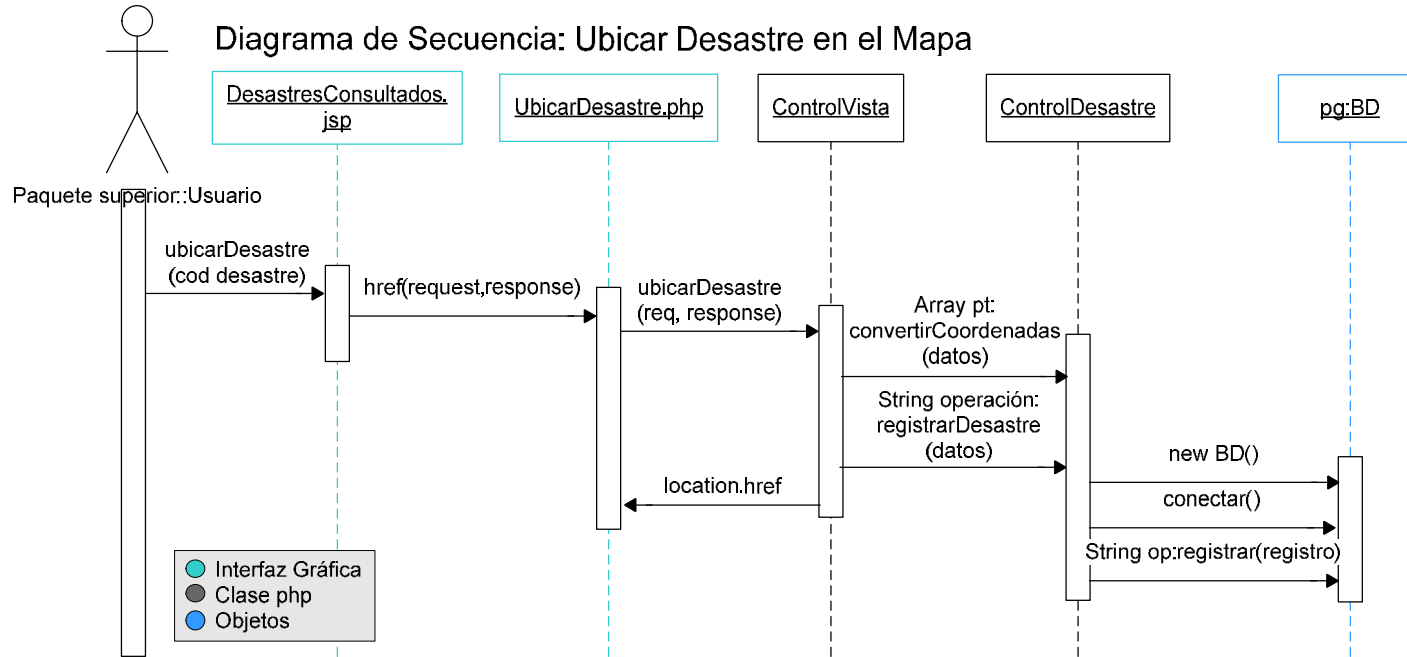
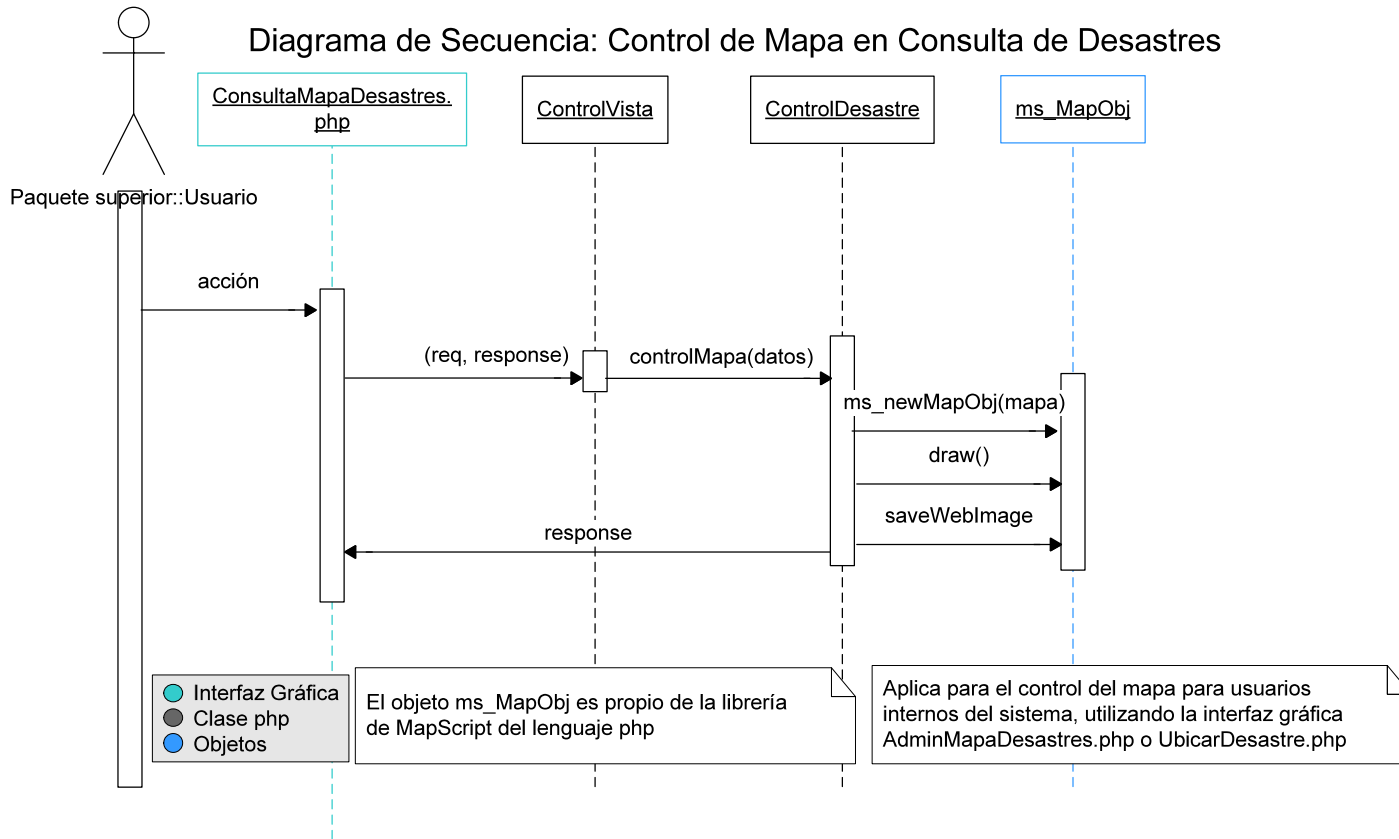


Diagrama de Secuencia: Control de Mapa en Consulta de Desastres



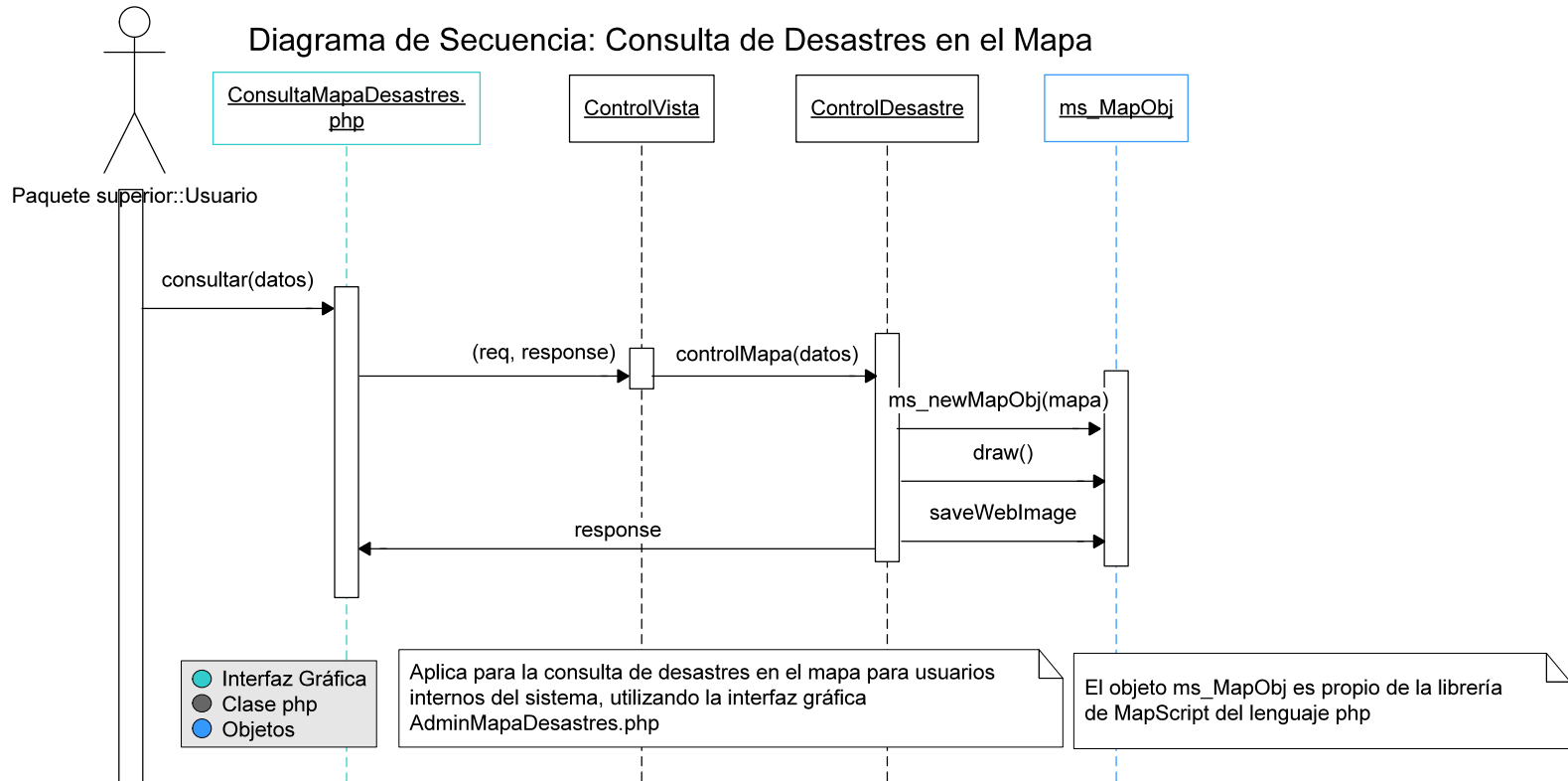
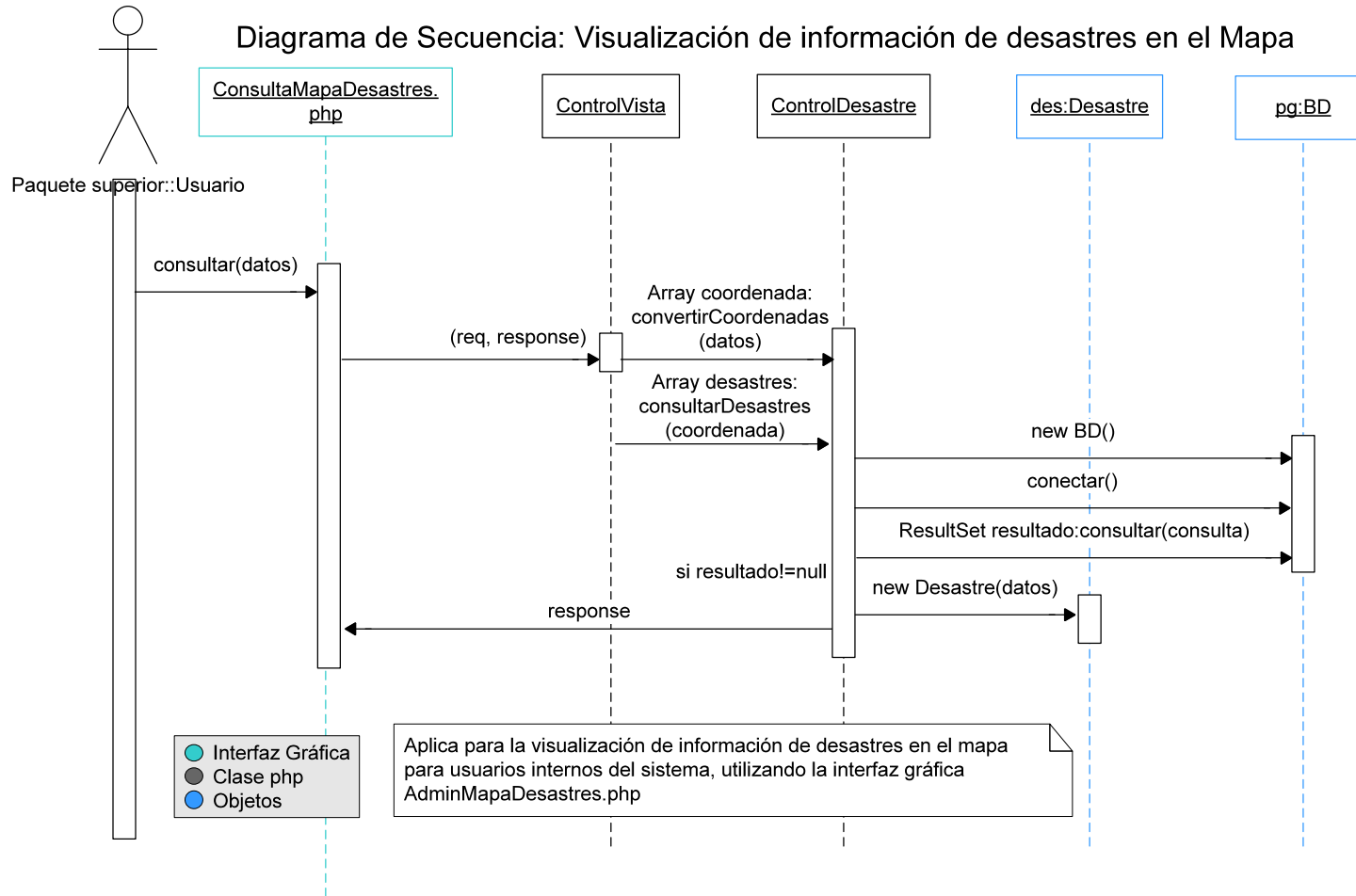


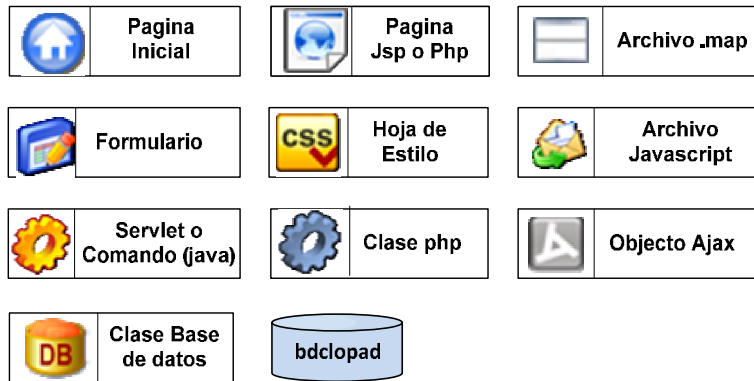
Diagrama de Secuencia: Visualización de información de desastres en el Mapa



7. MÓDELO DE DISEÑO

7.1 MODELOS DE NAVEGACIÓN

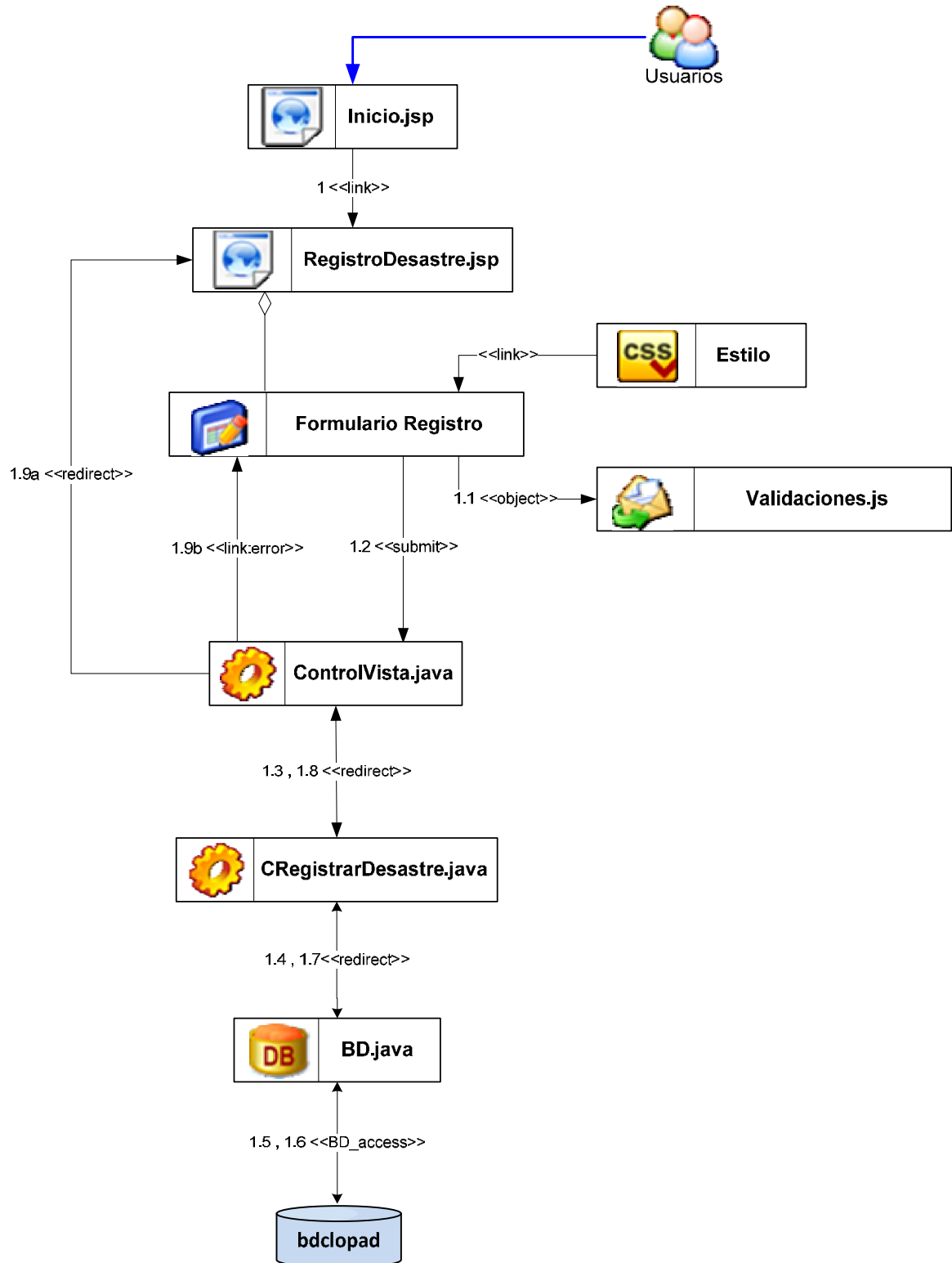
Esterotipos



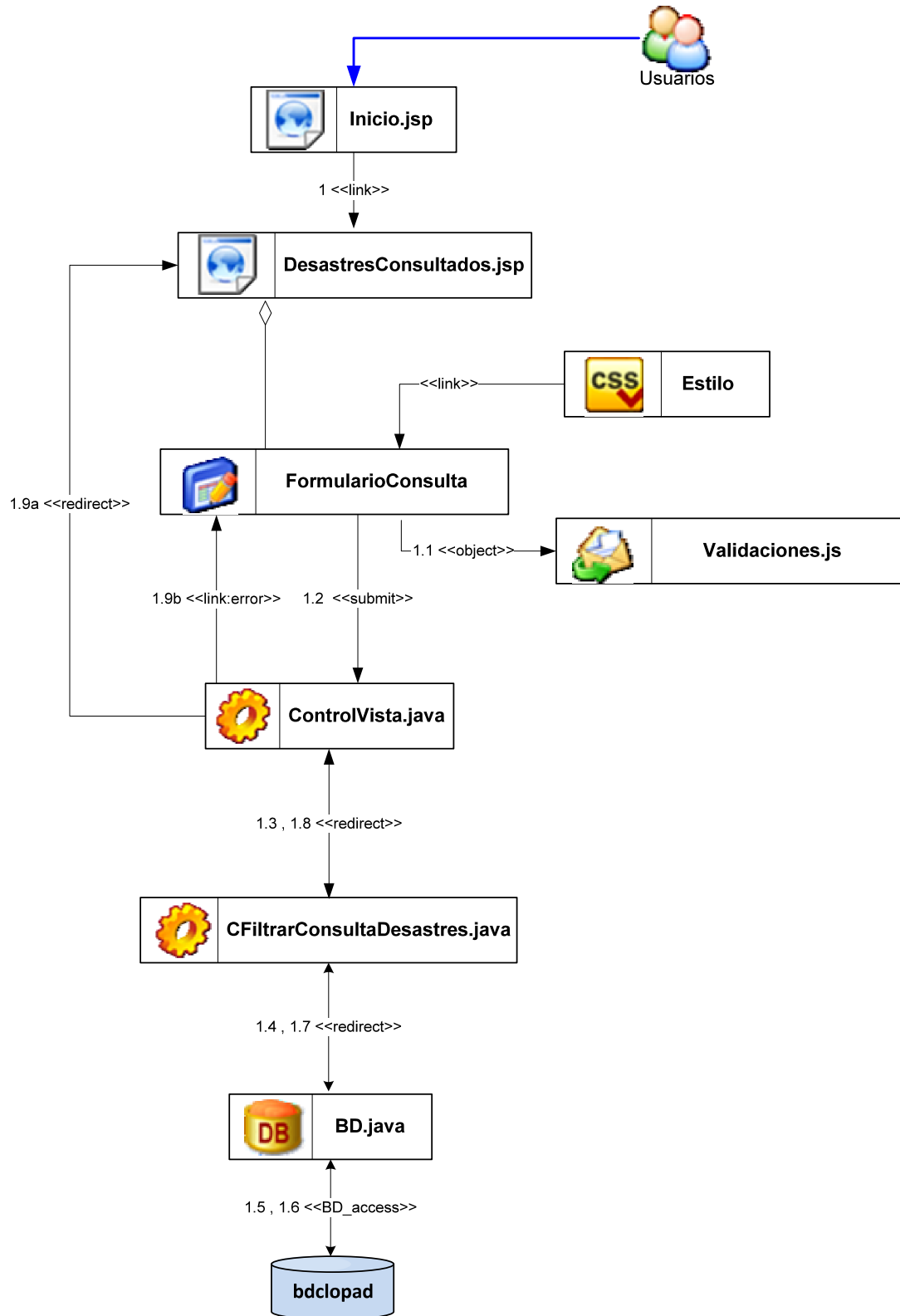
Asociaciones utilizadas en el modelo

- **<<link>>**: Se usa para establecer un enlace de navegación entre dos páginas del cliente, o de una página del cliente hacia una página del servidor.
- **<<build>>**: Se usa para representar la creación de una página del cliente desde una página del servidor, las páginas que son creadas de esta forma se llaman páginas de creación dinámica y las creamos en PHP.
- **<<redirect>>**: Se usa para relacionar dos o más páginas del servidor como en nuestro caso dos scripts de PHP.
- **<<submit>>**: Se usa para representar el envío de información de una página de formulario hacia una página del servidor.
- **<<target link>>**: Se usa para indicar enlaces entre páginas de un mismo Frame como en nuestro caso que trabajamos con IFrames.
- **<<object>>**: Se usa para relacionar objetos que cumplan una actividad específica dentro de una página. En nuestro caso, aplica para los javascript que validan los campos de un formulario.
- **<<DB_access>>**: Indica accesos o cualquier tipo de consulta a la base de datos.

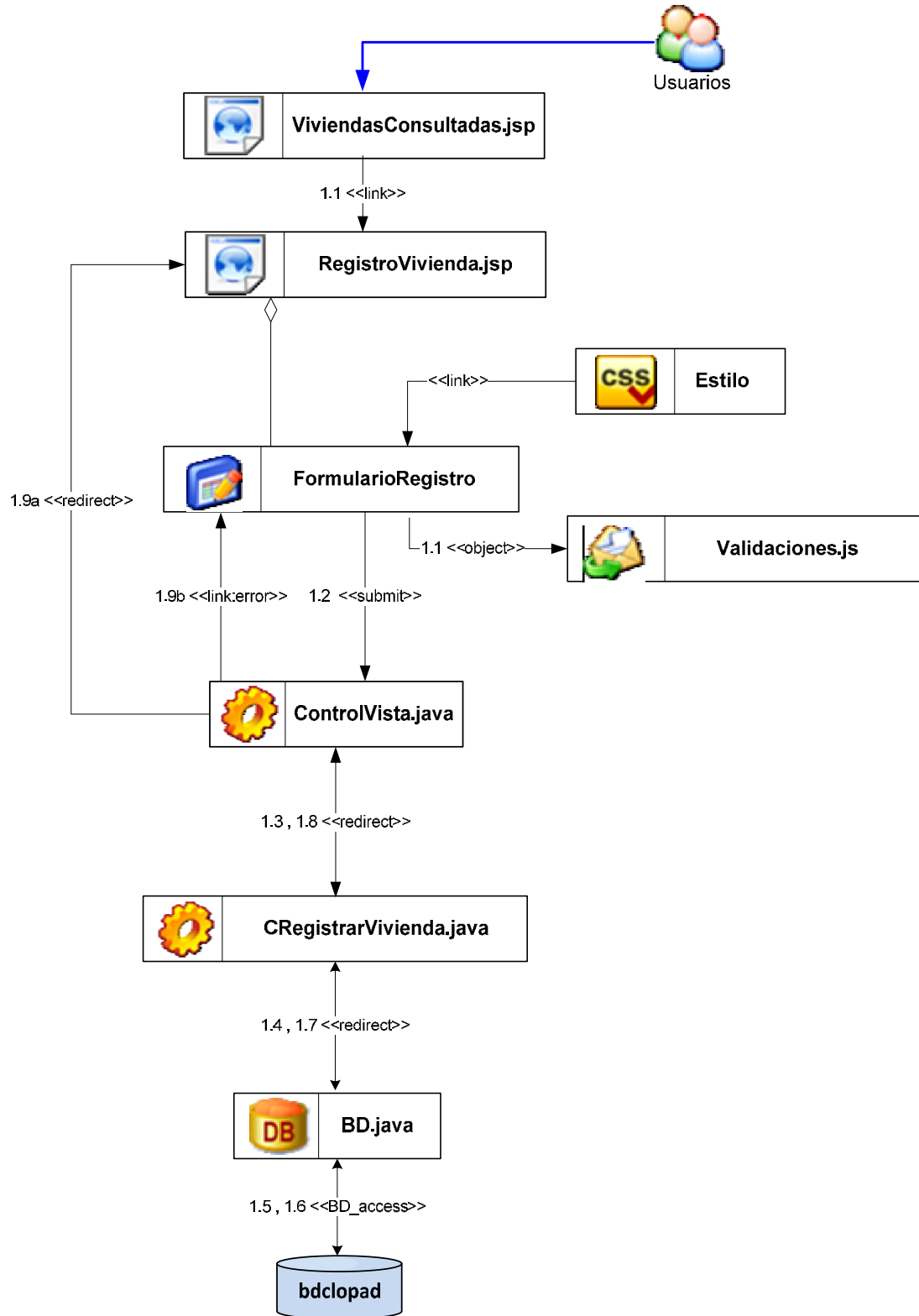
Registrar Desastre



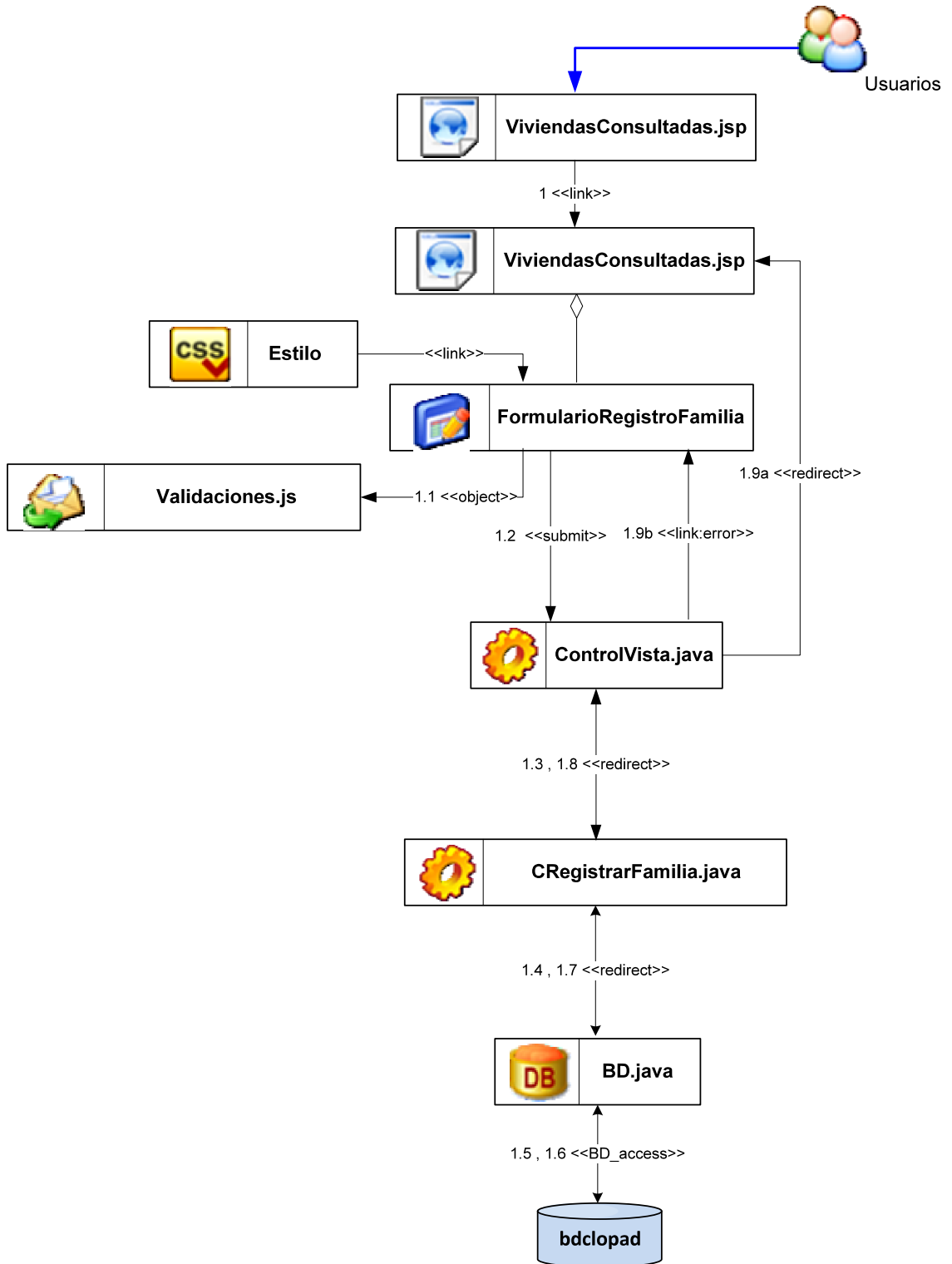
Consultar Desastres



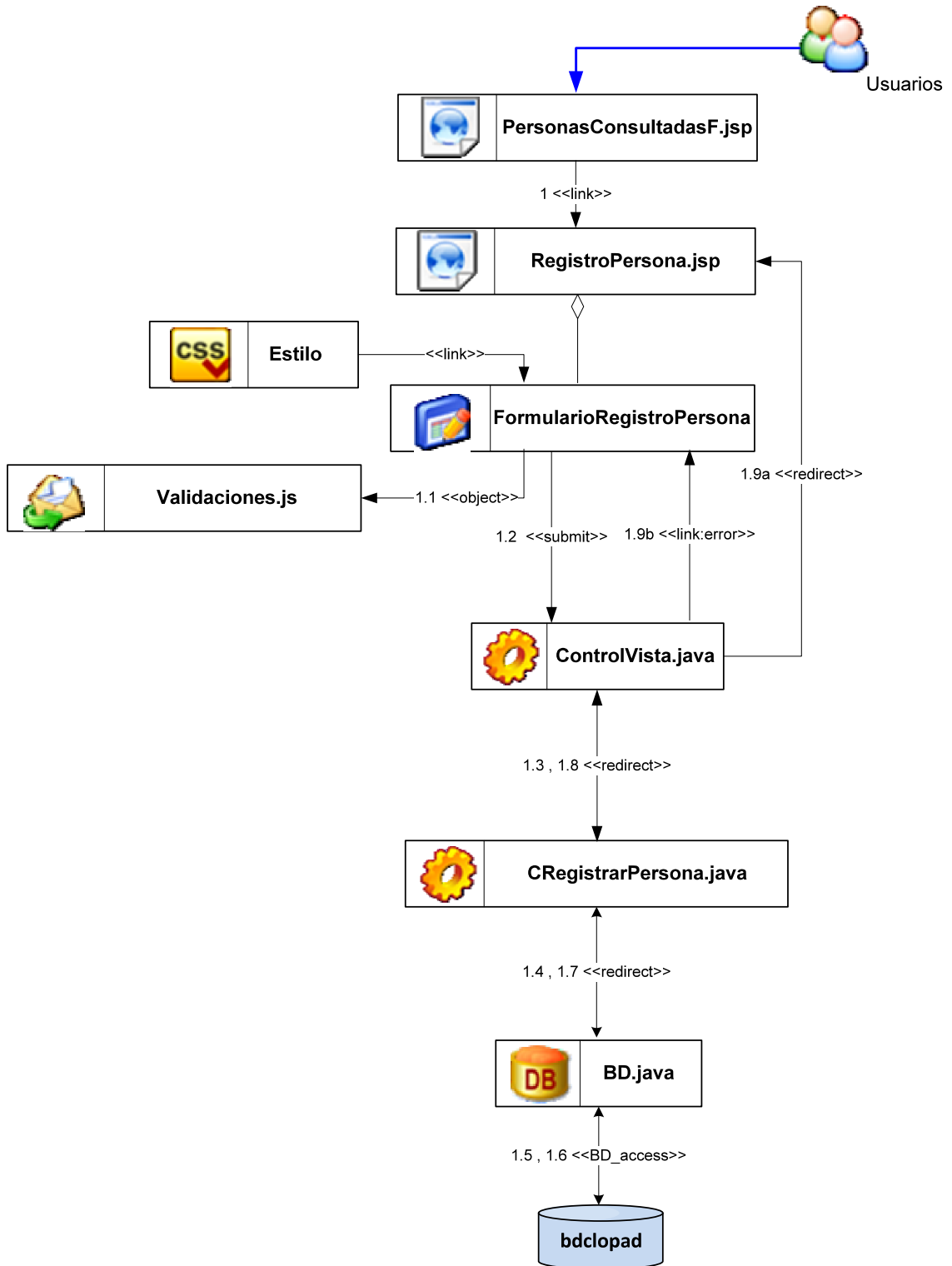
Registrar Vivienda



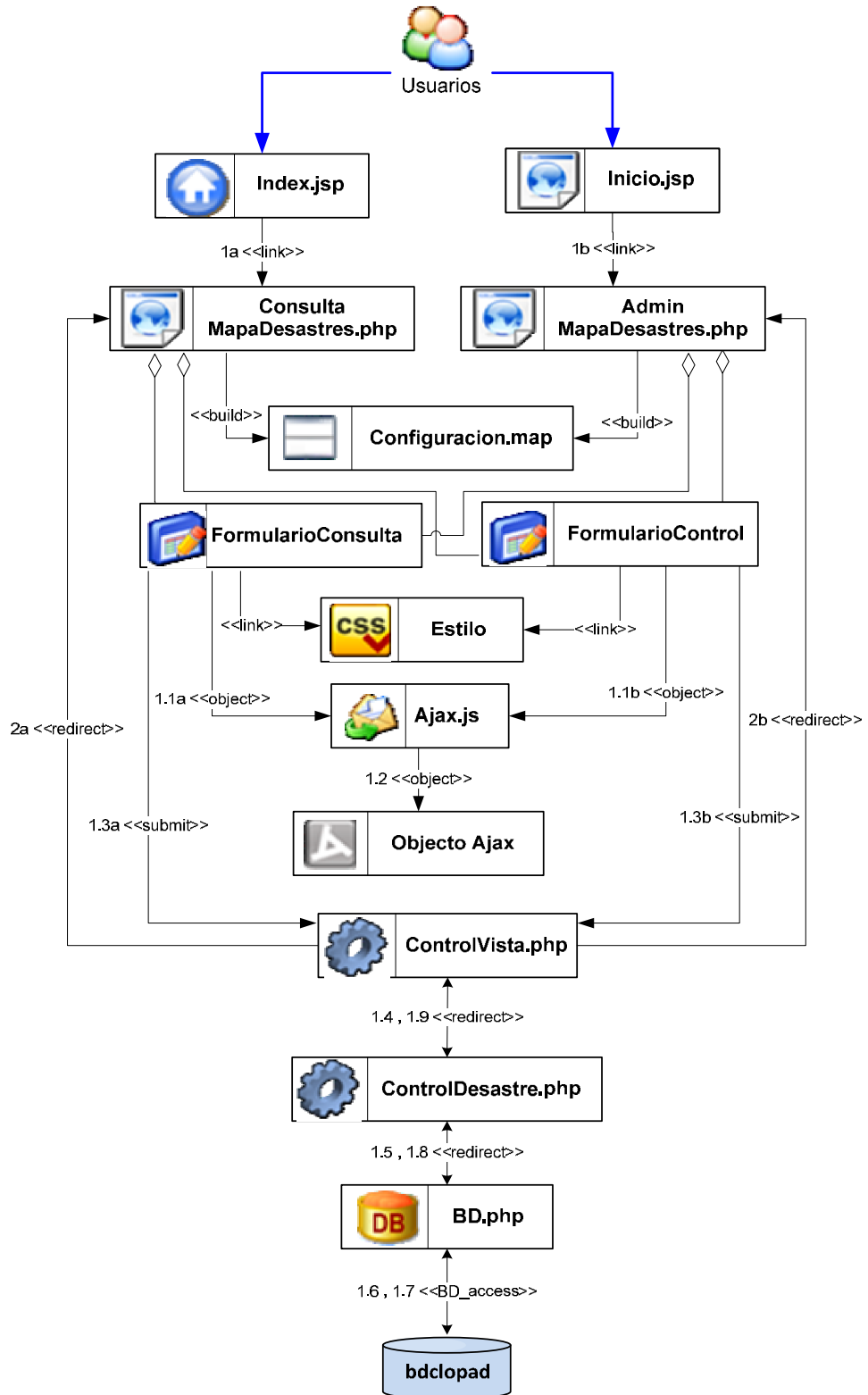
Registro Familia



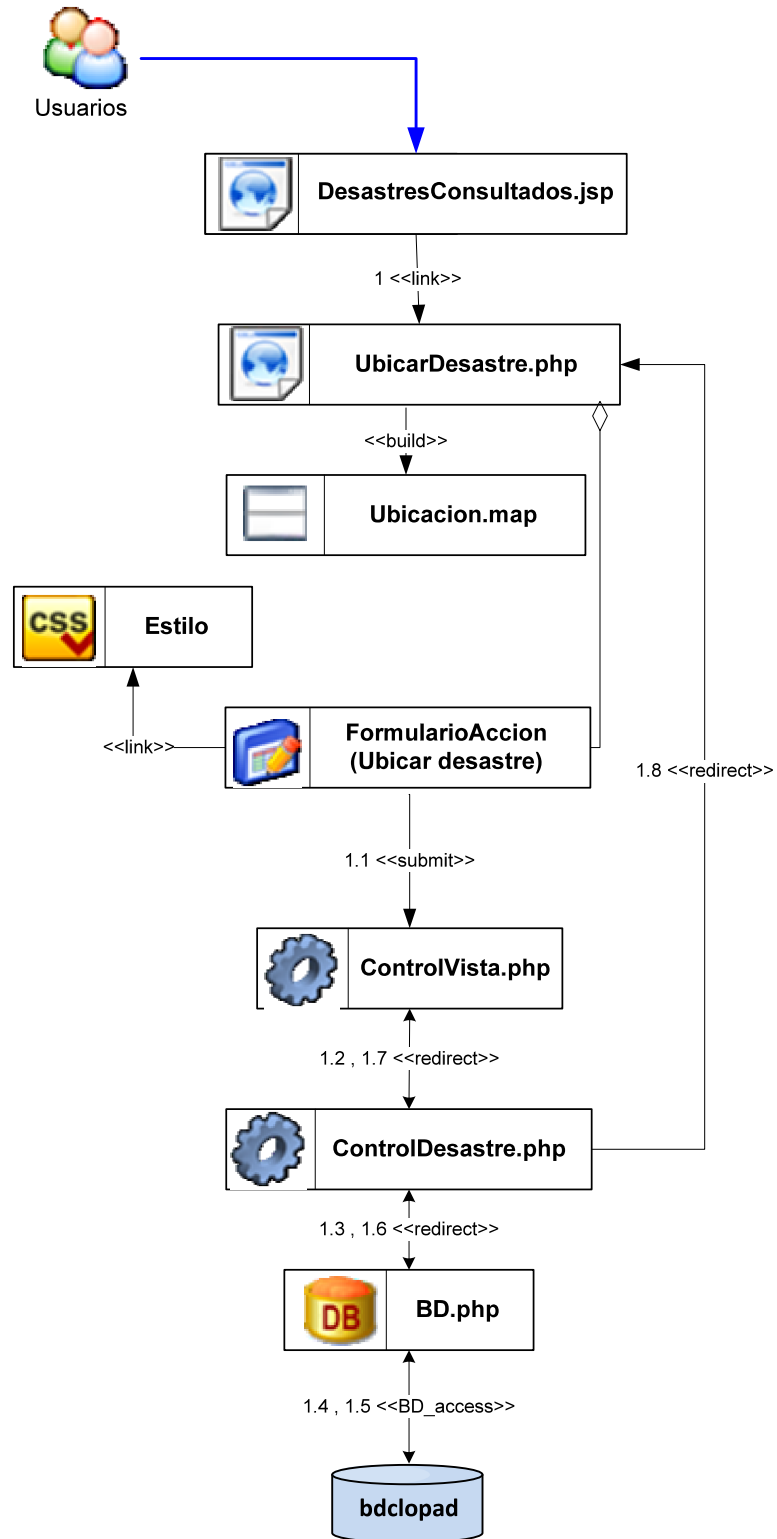
Registro Persona



Consulta de Desastres en el mapa



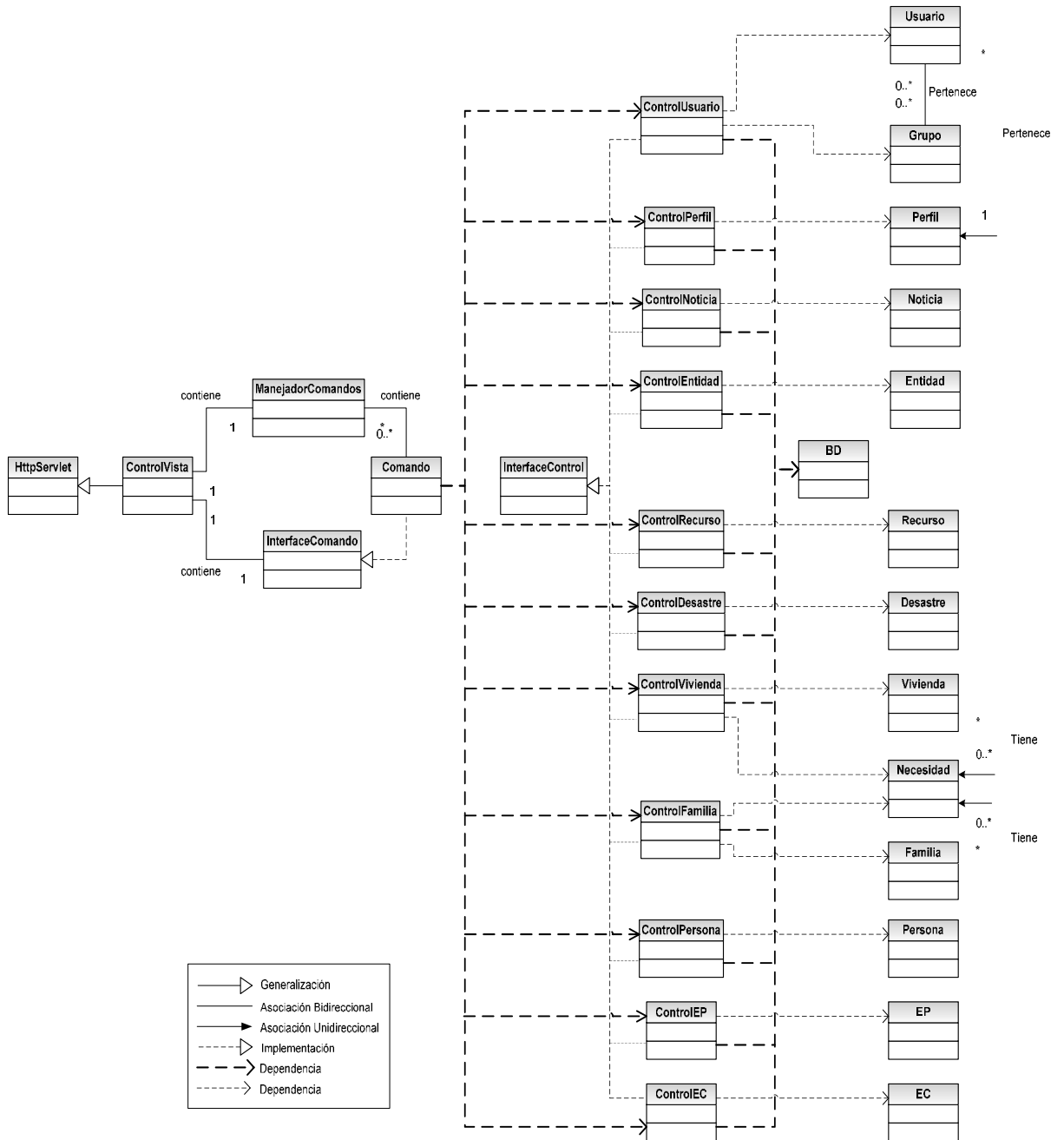
Ubicar desastre en Mapa



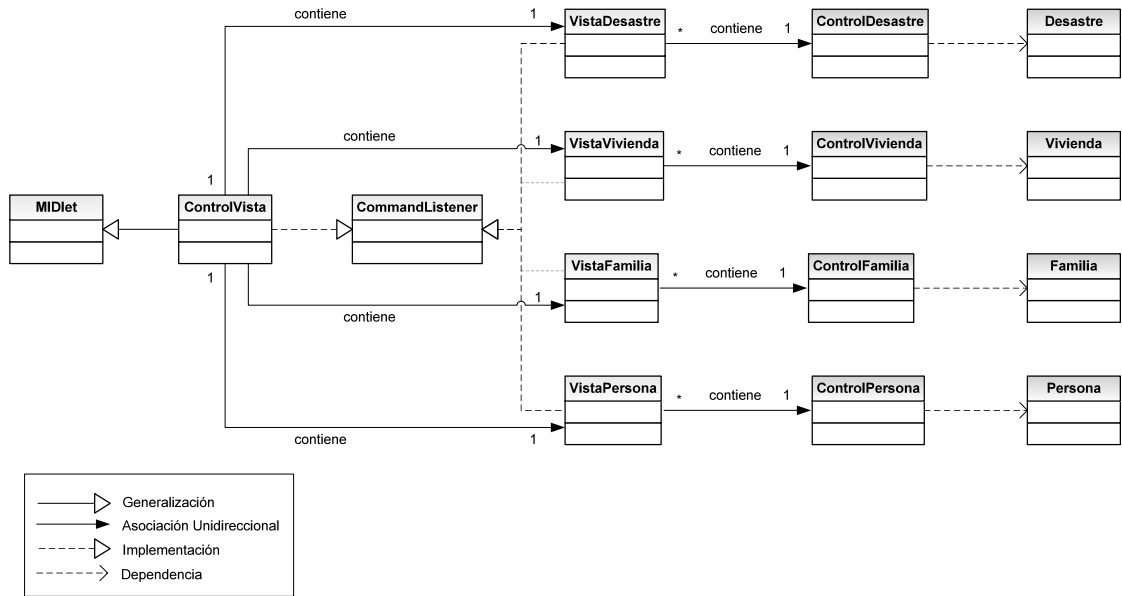
7.3 DISEÑO DE CLASES

7.3.1 Diagramas de las Clases

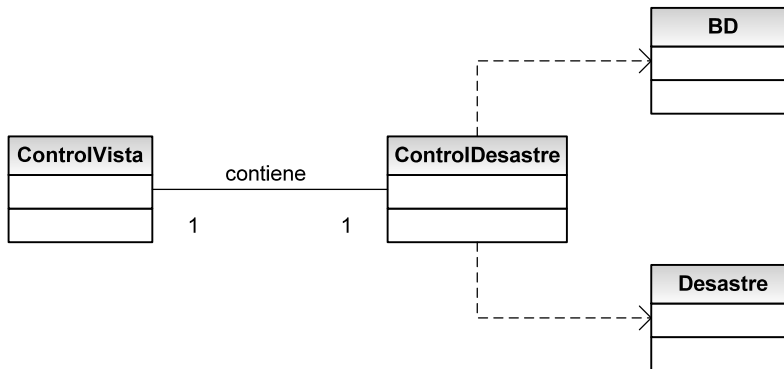
7.3.1.1 Aplicación General



7.3.1.2 Aplicación de Respaldo J2ME

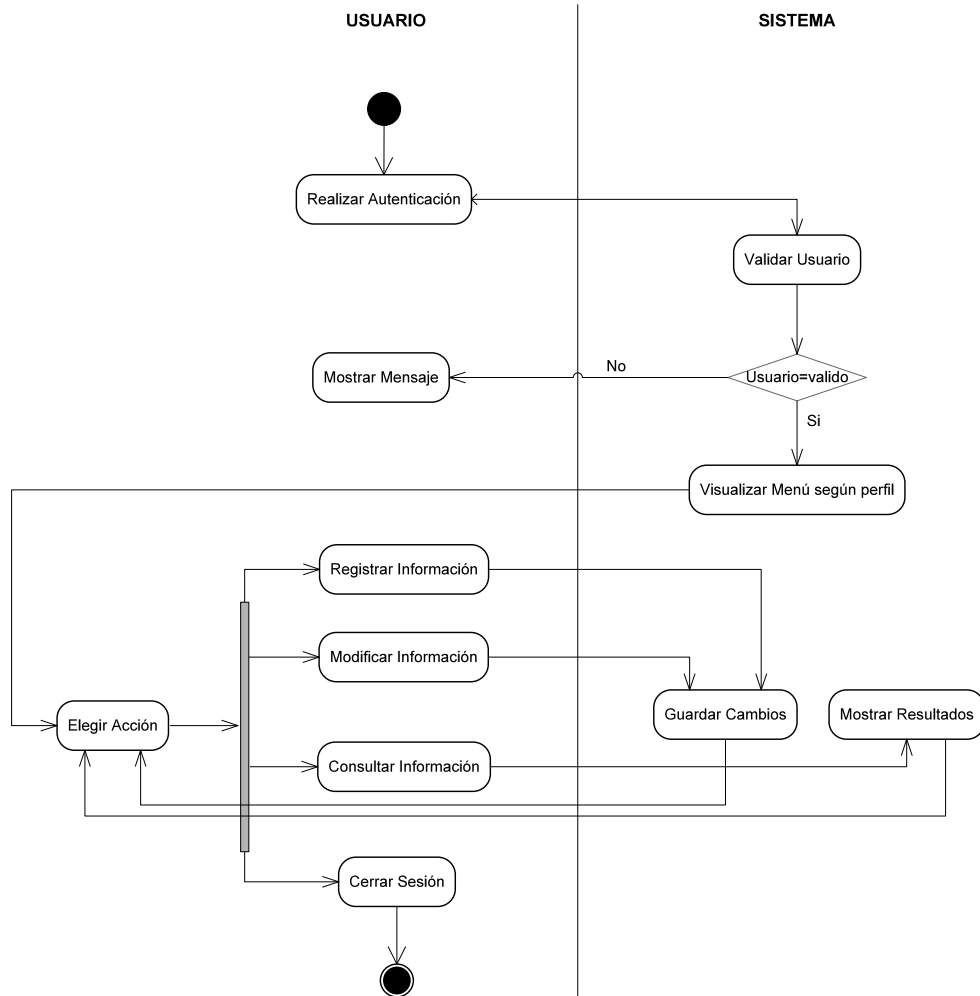


7.3.1.3 Aplicación Espacial

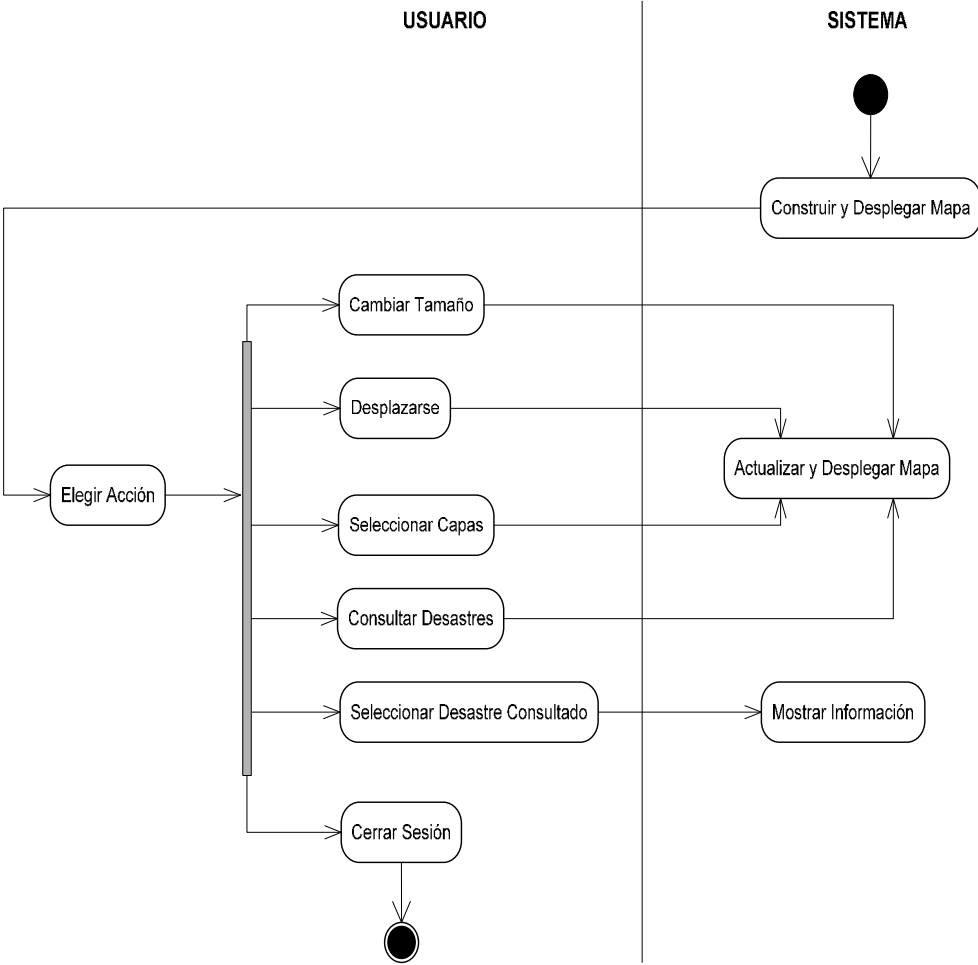


7.4 DIAGRAMA DE ACTIVIDADES

Diagrama de Actividad – Funciones del Sistema
Aplicación General y Móvil



**Diagrama de Actividad – Funciones del Sistema
Aplicación Espacial**



8. IMPLEMENTACIÓN

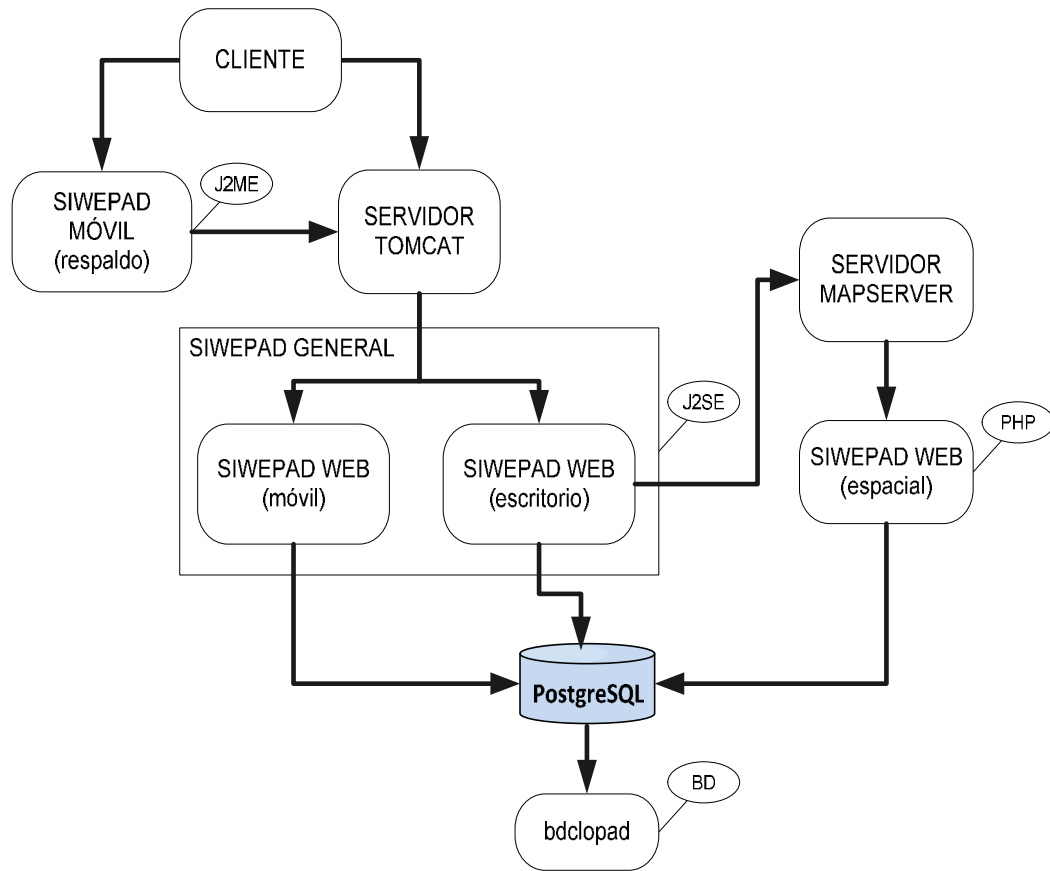


Figura 10. Sistema SIWEPAD

El Sistema de Información Web para la Prevención y Atención de Desastres (SIWEPAD) producto del desarrollo del trabajo de grado está compuesto por una aplicación Web General y una aplicación Web Espacial codificadas en lenguaje Java y PHP respectivamente, éstas utilizan una base de datos implementada en el motor PostgreSQL para el registro y recuperación de información; además de una aplicación de respaldo que permite la utilización de algunas funcionalidades del sistema, en dispositivos móviles, cuando no exista conexión a éste.

Dicho sistema está dirigido principalmente a los miembros del CLOPAD y de entidades pertenecientes a él, que se dividirán en los siguientes tipos de usuarios:

Administrador: Usuario que podrá registrar, consultar, modificar y eliminar datos de cualquier módulo de la aplicación.

Usuario Interno: Usuario que podrá registrar, consultar y modificar datos de los módulos que le sean permitidos de acuerdo al perfil que le haya sido asignado por el administrador.

Usuario Externo: Usuario que podrá consultar los datos concernientes a reportes e informes sobre los eventos ocurridos en la ciudad, igualmente podrá consultarlos en el mapa de Tuluá.

Una característica importante es que el usuario no tiene conexión directa con la aplicación espacial, para ello, él debe conectarse al Sistema y a través de la aplicación general conseguir conectarse a ésta. Igualmente, no es posible que la aplicación móvil de respaldo almacene datos directamente en la base de datos *bdclopap*, esto también se logra utilizando la aplicación general. Pero para un usuario los aspectos anteriores son transparentes, sólo se encarga de enviar peticiones o solicitudes al sistema y recibe la respuesta(s) de éste, sin saber qué elemento de él la(s) produjo.

Con el fin de facilitar el control en el mantenimiento del sistema, los elementos de las aplicaciones mencionadas se organizaron de acuerdo a la arquitectura de implementación MVC, que permite separar las interfaces gráficas, las clases que cumplen con las diversas actividades del sistema (Controles) y las clases modelos.

8.1 SIWEPAD GENERAL

Dispone de diferentes módulos, los cuales se manejan de acuerdo a las restricciones definidas por cada tipo de usuario, los módulos son los siguientes:

Módulo Usuarios: Módulo que permite registrar, consultar, modificar y eliminar la información correspondiente a usuarios, grupos de usuarios y perfiles.

Módulo Desastres: Módulo que brinda la posibilidad de registrar, consultar y actualizar información de desastres, viviendas, familias, personas, daños de vivienda y personas, necesidades de viviendas y familias, evaluación preliminar, evaluación complementaria. Además la generación de la evaluación final del desastre con su consolidado de daños cualitativo y cuantitativo y la determinación de necesidades.

Módulo Recursos: Módulo que ofrece las opciones de registrar, consultar, modificar o eliminar la información de los recursos con los que cuenta el CLOPAD para atender un desastre y las entidades que los proporcionan.

Módulo Reportes: Módulo que permite al administrador la generación de reportes parciales o totales de desastres en general, y de cada una de las evaluaciones al igual que del análisis de necesidades.

Módulo Configuración: Aplicación que ofrece la posibilidad de registrar información que permita la personalización de los formularios a utilizar en el levantamiento de información. Esta información está relacionada principalmente con la gestión de noticias, eventos o capacitaciones, y la configuración de elementos que puedan cambiar en el tiempo, como los tipos de desastres, vías de acceso, tipos de necesidades y recursos, etc. Lo que pretende este módulo es hacer la aplicación más configurable por el administrador, sin que en la eventualidad de un cambio en información definida se tenga que recurrir a modificaciones internas del sistema.

Su construcción está dividida en tres paquetes: Vista, Control y Modelo que corresponden a la arquitectura MVC, y un paquete adicional Almacenamiento que permite la conexión con la base de datos.

Modelo

En este paquete se encuentran todas las clases necesarias para representar la abstracción de los elementos que pertenecen al sistema real.

Cada clase modelo consta de atributos privados y de métodos públicos que permiten acceder a ellos ya sea para asignar u obtener su valor asociado.

Clase1
atributo1: tipoa
atributo2: tipoa
atributo3: tipob
...
asignarAtributo1(variable: tipoa): void
...
obtenerAtributo1(): tipoa
...

Figura 11. Elementos de Modelo SIWEPAD General

Vista

Compuesta por archivos JSP, páginas donde se combina código HTML y Java para generar contenido dinámico, que representan la interfaz gráfica de la aplicación; una clase *ControlVista* que hereda de la clase *Servlet* donde se reciben los datos ingresados por un usuario y la acción que se desea ejecutar, lo que se almacena en el objeto *request*; de allí, se direcciona a una clase *ManejadorComandos* que, de acuerdo a la acción, se encarga de generar el comando que corresponde a la petición recibida, el cual realiza la recuperación de los datos enviados para la ejecución de las funciones necesarias que dan cumplimiento a la acción indicada. El direccionamiento desde *ControlVista* hasta un comando determinado se logra a través de una única interfaz *InterfaceComando* que es implementada por el comando indicado.

La clase *ManejadorComandos*, la interfaz *InterfaceComando* y las diferentes clases *Comandos* asociadas, representan la implementación del patrón de diseño por comandos.

La utilización de este patrón tiene varias ventajas según [Castejón, 2004]: permite cambiar fácilmente el procesamiento para una acción determinada sustituyendo el comando que la implementa por otro, permite reutilizar comandos y favorece el encadenamiento de dos o más comandos para la implementación de tareas complejas.

Teniendo en cuenta estas consideraciones, se escogió este patrón como herramienta en la implementación del proyecto.

Igualmente, en la construcción y en el manejo de la interfaz gráfica se utilizaron herramientas que facilitan el diseño y permiten validar y enviar de manera eficiente los datos ingresados a través de ella. Estas herramientas son: hojas de estilo en cascada, archivos *javaScript* y la tecnología *Ajax*.

Para dar extensión de algunas funcionalidades de la aplicación hasta dispositivos móviles conectados a ésta, se crearon archivos WML que contienen contenido estático, y se construyeron comandos que generan este tipo de archivos con contenido dinámico en tiempo de ejecución. La ejecución de las funcionalidades por medio de estos dispositivos se realiza de igual manera que en computadores de escritorio; se reciben datos y una acción que activa el comando, dentro del cual se efectúan las tareas que cumplen con la funcionalidad requerida.

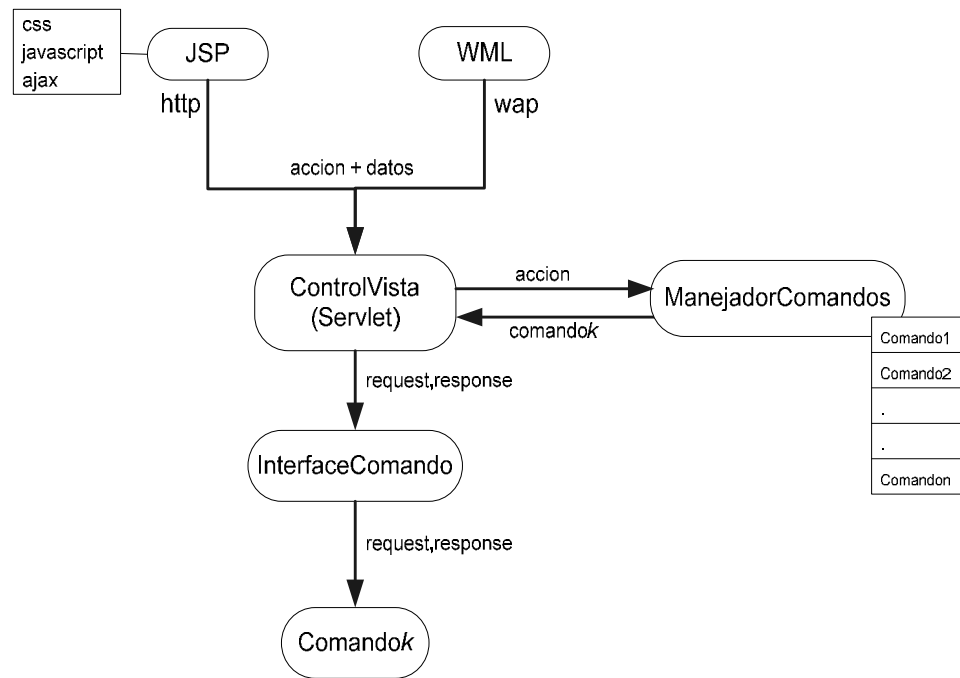


Figura 12. Elementos de Vista SIWEPAD General

Control

El paquete Control contiene las clases *Controles* y una interfaz *InterfaceControl* por cada una de ellas. Cada control contiene los métodos necesarios para gestionar uno o varios elementos del modelo.

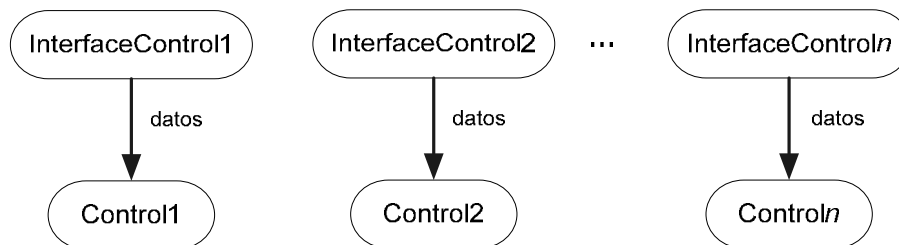


Figura 13. Elementos de Control SIWEPAD General

Almacenamiento

Consta de una sola clase *BD* que contiene métodos que permiten establecer conexión con la base de datos *bdclopap* a través del sistema manejador de base de datos PostgreSQL, para ingresar y consultar información en ella. Los métodos generados son de tipo *static*, lo que brinda la posibilidad de utilizarlos sin necesidad de crear una instancia de la clase.



Figura 14. Elementos de Almacenamiento SIWEPAD General

Para controlar las actividades que un usuario puede ejecutar, se da la posibilidad de generar perfiles que limiten el acceso a los diferentes módulos y acciones de la aplicación. Lo anterior permite que, una vez ingrese un usuario a la aplicación se genere su menú dinámicamente de acuerdo al perfil que tenga asignado, además se crea una sesión mediante la clase *HttpSession*, que permite controlar su navegación por la aplicación.

8.2 SIWEPAD ESPACIAL

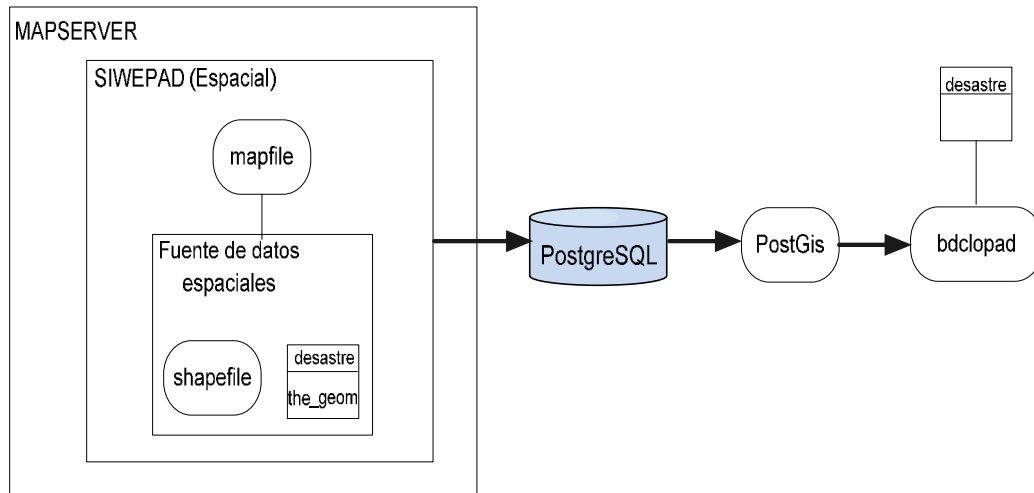


Figura 15. Arquitectura SIWEPAD Espacial

Su construcción puede resumirse en la articulación de diversas herramientas que permitieron generar una aplicación que incorpora una referencia espacial a información de desastres obtenida mediante la aplicación General, y que además permite visualizar dicha información a través de un mapa.

La aplicación Web codificada en lenguaje PHP utiliza archivos *mapfile* para dar formato a los mapas que en ella se visualizan, estos archivos utilizan una fuente de datos espaciales conformada por una columna Geométrica de la tabla desastre de *bdclpad* y por archivos *shapefile* proporcionados por la oficina de Planeación Municipal de la Alcaldía de Tuluá.

El funcionamiento de la aplicación es similar al de la aplicación general, se reciben datos y una acción en un archivo *ControlVista* que invoca métodos de *ControlDesastre* según la funcionalidad que el usuario desee realizar.

8.2.1 Visualización de Mapas

Para visualizar información espacial en forma de mapa es necesario crear una aplicación Web que en su interfaz gráfica incluya una porción de código php donde, utilizando la librería *mapscript*, se cree un objeto de la clase *objMap* con base en la información de un archivo *mapfile* (Ver Anexo -) configurado previamente, por medio de éste se crea y almacena la imagen y, utilizando la ruta donde fue almacenada se visualiza con alguna etiqueta del código HTML que lo permita.

8.2.2 Manipulación del Mapa

La manipulación consiste en cambiar los atributos o la forma cómo se está visualizando un determinado mapa, por ejemplo hacer un aumento o disminución de éste, aclarando que estas operaciones no modifican las dimensiones de la región de la interfaz sobre la que se aloja el mapa o imagen.

Para lograr esto es también necesario hacer uso de la clase *objMap* consiguiendo por medio de las acciones del usuario los datos o parámetros requeridos en los distintos métodos que la clase ofrece. En éste orden de ideas, cada vez que un usuario ejecuta una acción en la interfaz se capturan y se envían los datos hacia *ControlDesastre* donde se procesan y se produce un nuevo mapa.

Debido a que el mapa es visualizado sólo en una porción de la interfaz y es sólo ésta parte la que cambia de contenido cada vez que se ejecute una acción, fue útil el uso de Ajax, para actualizar sólo la región del mapa y conservar el contenido restante, esto para cuestiones de eficiencia en el envío de datos hacia la aplicación.

El archivo *mapfile* ofrece una configuración inicial para el mapa, pero es posible cambiar esta configuración dinámicamente dentro de la aplicación, que en efecto fue la esencia del trabajo hecho en lo que a manipulación de mapas se refiere.

8.3 SIWEPAD MÓVIL DE RESPALDO

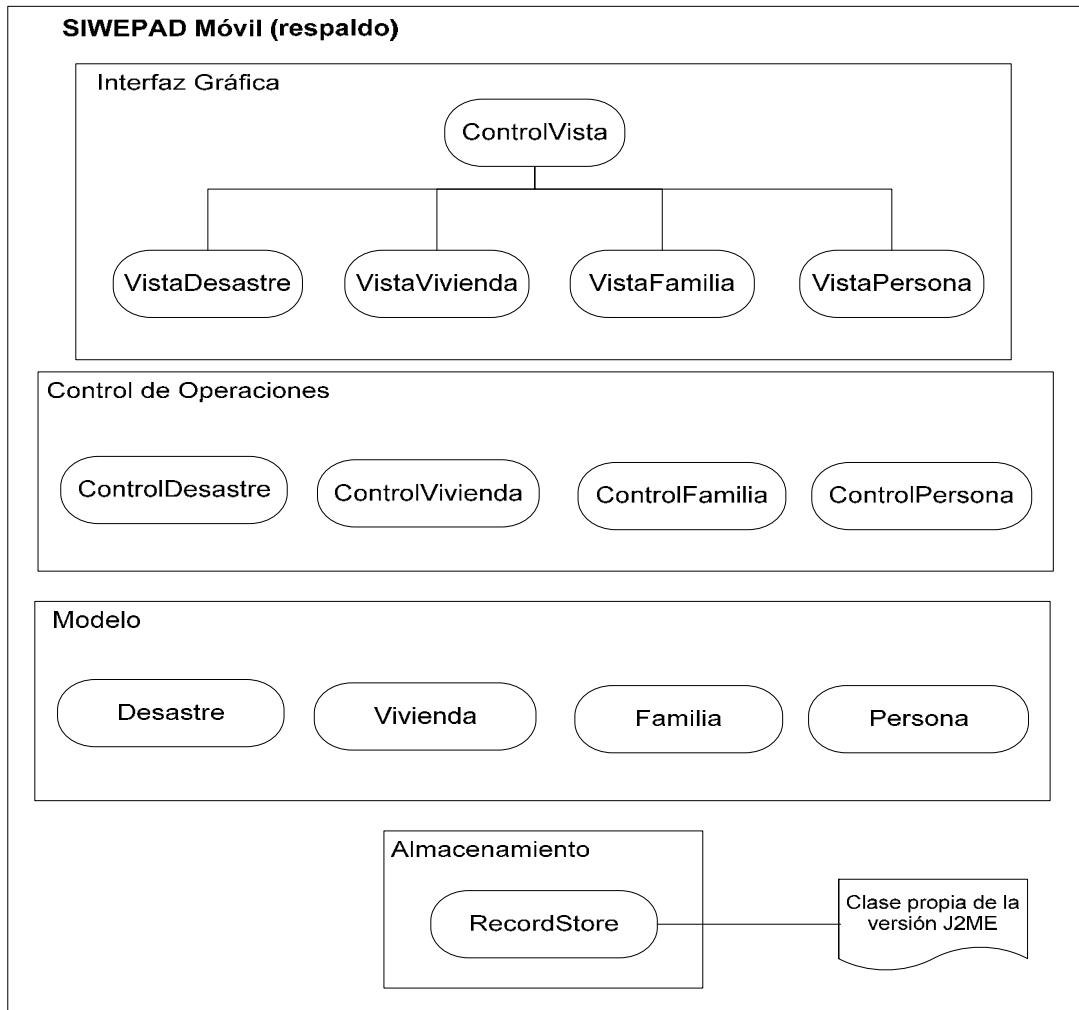


Figura 16. Componentes principales Arquitectura SIWEPAD Móvil

Su construcción fue hecha bajo la edición J2ME de Java. Se creó una clase *ControlVista* que, hereda de la clase *MIDlet*, implementa la clase *ActionListener* y se encarga del manejo de la interfaz gráfica. Para el manejo de información en cada operación se utilizan clases que tienen como función generar los formularios y demás elementos de interfaz que son retornados a *ControlVista* necesarios para mostrar una determinada información.

Almacenamiento de información

El almacenamiento persistente de datos en la memoria del dispositivo se logra gracias a la clase *RecordStore*, teniendo también esta aplicación la posibilidad de exportar esa información hacia la aplicación general para ser almacenados en la base de datos *bdclpad* mediante el envío de estos utilizando el protocolo *WAP*.

9. PRUEBAS Y /O RESULTADOS

9.1 ÁMBITO

Contiene las pruebas funcionales, técnicas y de integridad realizadas al sistema desarrollado, para garantizar que funcione correctamente, y que genere los datos correctos según las entradas proporcionadas, además de garantizar el buen funcionamiento en situaciones anormales.

Los principales aspectos a evaluar son:

Pruebas de Caja Negra


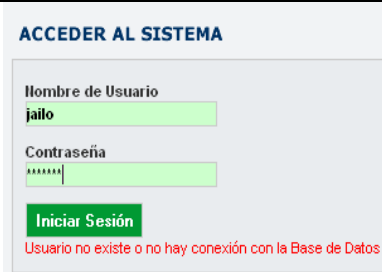
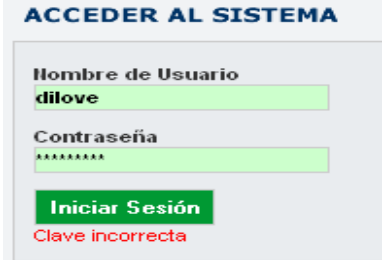
- Autenticación de Usuarios
- Validación de información ingresada por los usuarios
- Despliegue de mensajes de error o de confirmación
- Duplicidad de información
- Integridad de datos
- Envío y recepción de información entre la aplicación móvil y el servidor
- Despliegue de ubicación de desastres en el mapa
- Despliegue de información de desastres
- Funcionamiento del control del mapa

Pruebas de Caja Blanca

Estas pruebas fueron realizadas a medida que se codificaban las diferentes funcionalidades del sistema. Para ello se utilizó el despliegue interno de mensajes en la consola del editor, esto permitió llevar a cabo el seguimiento de las diferentes funciones ejecutadas. Las pruebas más importantes que se realizaron son las siguientes:



- Manejo de Excepciones
- Utilización adecuada de las estructuras de control
- Generación de porciones de código reutilizables
- Definición correcta de condiciones de parada de ciclos
- Conexión a Base de Datos

9.2 PROCEDIMIENTO Y RESULTADOS OBTENIDOS

NOMBRE	Autenticación de Usuarios	
PROPÓSITO	Validar si el sistema solo permite el ingreso de usuarios registrados con la información correcta.	
Ejemplo:		
DATOS DE ENTRADA	RESULTADOS ESPERADOS	DATOS DE SALIDA
Usuario: hermanvera Contraseña: herman	Ingreso Exitoso	 <p>SIWEPAD Sistema de información Web Para La Pr...</p> <p>SIWEPAD / Inicio</p> <p>Inicio Administrar Desastres Administrar Recursos</p> <p>SIWEPAD / Inicio - Bienvenido Herman Alfonso</p> <p>Sistema de Información Web Para el CLOPAD</p>
Usuario: jailo Contraseña: ja1201	Usuario no Existe	 <p>ACCEDER AL SISTEMA</p> <p>Nombre de Usuario jailo</p> <p>Contraseña *****</p> <p>Iniciar Sesión</p> <p>Usuario no existe o no hay conexión con la Base de Datos</p>
Usuario: dilove Contraseña: dianalore	Clave Incorrecta	 <p>ACCEDER AL SISTEMA</p> <p>Nombre de Usuario dilove</p> <p>Contraseña *****</p> <p>Iniciar Sesión</p> <p>Clave incorrecta</p>

NOMBRE	Validación de información ingresada por los usuarios	
PROPÓSITO	Probar si la aplicación valida la información ingresada, como campos vacíos o datos incorrectos.	
Ejemplo:		
DATOS DE ENTRADA	RESULTADOS ESPERADOS	DATOS DE SALIDA
Nombre: Inundación Fecha: 2008-07-25 Hora de ocurrencia: 03:00:00 Area/Zona: Cienegueta Tipo de Evento: Natural Impacto: Nivel I	Registro Exitoso	<p style="text-align: center;">* Ingreso Obligatorio</p> <p style="text-align: center;">Desastre registrado exitosamente</p>
Nombre: Fecha: 2008-07-25 Hora de ocurrencia: 03:00:00 Area/Zona: Cienegueta Tipo de Evento: Natural Impacto: Nivel I	El campo nombre está incompleto	<p style="text-align: center;">* Ingreso Obligatorio</p> <p style="text-align: center;">El campo nombre tiene información incompleta</p> <p>Información de Registro</p> <p>Fecha de Registro <input type="text" value="2008-07-26"/></p> <p>Hora de Registro <input type="text" value="18:25:32"/></p> <p>Codigo <input type="text" value="16"/></p> <p>*Nombre <input style="border: 1px solid red;" type="text"/></p>
Nombre: Inundación Fecha: 2008-07-25 Hora de ocurrencia: 56:00:00 Area/Zona: Cienegueta Tipo de Evento: Natural Impacto: Nivel I	La hora de ocurrencia es incorrecta	<p style="text-align: center;">La hora de ocurrencia ingresada est incorrecta</p> <p>Información de Registro</p> <p>Fecha de Registro <input type="text" value="2008-07-26"/></p> <p>Hora de Registro <input type="text" value="18:30:31"/></p> <p>Codigo <input type="text" value="16"/></p> <p>*Nombre <input type="text" value="Inundacion"/></p> <p>Información de Ocurrencia</p> <p>*Fecha <input type="text" value="2008-07-25"/></p> <p>Hora de Ocurrencia <input style="border: 1px solid red;" type="text" value="56:00:00"/> hh:mm:ss (formato 24h)</p>

NOMBRE	Duplicidad de información	
PROPÓSITO	Validar si el sistema indica al usuario cuando va a registrar información con un identificador ya existente.	
Ejemplo:		
DATOS DE ENTRADA	RESULTADOS ESPERADOS	DATOS DE SALIDA
Cedula: 1116235891 Nombre: Lorena Apellido: Vanegas Usuario: diana Clave: diana	Ya existe un usuario con esa cédula	<p style="text-align: center;">Ya existe un usuario con esa cedula</p> <p>Información Personal</p> <p>*Cédula <input type="text" value="1116235891"/></p>

NOMBRE	Integridad de Datos													
PROPÓSITO	Probar que la aplicación valida la información a registrar o actualizar, para garantizar la integridad de los datos almacenados, en acciones que genere por ejemplo datos no válidos a la base de datos o se puedan modificar datos existentes tomando un valor incorrecto.													
Ejemplo: Probar que el sistema solo deje registrar una familia por vivienda.														
DATOS DE ENTRADA	RESULTADOS ESPERADOS	DATOS DE SALIDA												
Código:1 Corregimiento: La Marina Zona: Urbana Tenencia: Arrendatario Familia: False	Opción Registrar Familia	 <table border="1"> <thead> <tr> <th>Código</th> <th>Corregimiento</th> <th>Familia</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>La Marina</td> <td></td> </tr> <tr> <td>2</td> <td>Barragan</td> <td> Registrar Familia</td> </tr> <tr> <td>3</td> <td>rio</td> <td></td> </tr> </tbody> </table>	Código	Corregimiento	Familia	1	La Marina		2	Barragan	Registrar Familia	3	rio	
Código	Corregimiento	Familia												
1	La Marina													
2	Barragan	Registrar Familia												
3	rio													
Código:10 Corregimiento: Bolivar Zona: Urbana Tenencia: Poseedor Familia: True	Opción Actualizar Familia	 <table border="1"> <tbody> <tr> <td>9</td> <td>aguita</td> <td></td> </tr> <tr> <td>10</td> <td>Bolivar</td> <td></td> </tr> <tr> <td>11</td> <td>valle</td> <td></td> </tr> <tr> <td>12</td> <td>vallecito</td> <td> Actualizar Familia</td> </tr> </tbody> </table>	9	aguita		10	Bolivar		11	valle		12	vallecito	Actualizar Familia
9	aguita													
10	Bolivar													
11	valle													
12	vallecito	Actualizar Familia												

RECOMENDACIONES

- La arquitectura de implementación adoptada en el desarrollo de la aplicación general, permite dar al sistema mantenibilidad y portabilidad, sin embargo no se logró implementar una arquitectura MVC pura, debido a que no se utilizó en el modelo el patrón de diseño *Colecciones* para el manejo y control en el almacenamiento de la información.
- El continuo mejoramiento del sistema es de vital importancia para garantizar el uso constante de éste y su utilidad en los procesos de prevención y atención de desastres.
- En la aplicación espacial, cada vez que se realiza una modificación al mapa, ésta genera una imagen en formato .jpg que es almacenada en el servidor en la siguiente ubicación C:\ms4w\Apache\htdocs\SiwepadEspacial\Vista\mapas, por lo tanto es conveniente que éstas sean eliminadas periódicamente para no ocupar espacio en disco duro con elementos innecesarios.

PROYECCIONES

El sistema desarrollado es una herramienta de gran utilidad en todos los procesos de atención, sin embargo los eventos de desastre generan cada vez mayores consecuencias, así que con el fin de mejorar la calidad de atención a personas afectadas por este tipo de evento se plantea incorporar al sistema las siguientes funcionalidades:

- Implementar un módulo de Prevención con el fin de extender del sistema hacia áreas como la gestión del riesgo y el manejo de información a nivel educativo.
- Construir de un Data WareHouse y aplicar técnicas de Descubrimiento de conocimiento con información concerniente a las áreas más relevantes del comité, que brinde una herramienta más completa para la toma de decisiones.
- Adecuar el sistema mediante la incorporación de módulos con el fin de extender su uso hacia entidades nacionales.
- Adaptar el sistema haciendo uso de los Servicios Web y del protocolo SOAP para permitir la cooperación y la interoperabilidad entre diferentes aplicaciones que manejen información relacionada con la prevención y atención de desastres.
- Incorporar al sistema un buscador que a partir de la Web Semántica, ofrezca de forma más rápida y sencilla a los usuarios, resultados más precisos en el proceso de consulta de información.

CONCLUSIONES

- Se desarrolló una aplicación general que permite la gestión de información en la etapa de atención de desastres, basado en los procesos actuales de recolección de datos y en algunos estándares de Evaluación de Daños y Análisis de Necesidades (EDAN) utilizados en el entorno colombiano.
- Se incorporó exitosamente el componente cartográfico operacional bajo la web que permite consultar la ubicación de los desastres visualizándolos en un mapa y la información descriptiva de los mismos.
- Se brindó la posibilidad de registrar y consultar la información más importante en cuanto al manejo de desastres, mediante dispositivos móviles, por medio de dos aplicaciones, una tipo web y otra cuando no se tenga comunicación directa con el servidor.
- Este sistema apoya el mejoramiento de los procesos llevados a cabo en la atención de desastres, siendo ésta una temática tan importante en la actualidad, que contribuye a mejorar la forma de atender a la población, bienes y servicios afectados por estos.
- Se investigaron y aplicaron herramientas para el manejo de información espacial, la cual es un componente importante del sistema, que sigue las nuevas tendencias de los sistemas informáticos.
- Se aplicaron conocimientos de desarrollo de aplicaciones móviles utilizando la versión J2ME y el lenguaje WML, esto permitió dar al sistema una característica vanguardista con respecto a incorporación de información a los sistemas web, a través del uso ágil y funcional de dispositivos móviles.
- La construcción del sistema se llevo a cabo mediante la utilización de herramientas de libre distribución, lo que minimizó los costos de implementación del proyecto y evita la realización de gestiones de adquisición de licencias en el momento que se desee implantar.

BIBLIOGRAFÍA

- [1] ABAD, Rodolfo Perez. Arquitectura Modelo/Vista/Controlador. Escuela Superior de Cómputo del Instituto Politecnico Nacional. Disponible en: <http://abad.galeon.com/mvc.pdf>. [Consultado en Febrero 2008]
- [2] BAZ I, KARAS I. R, GEYMEN A. AKAY A. E. *Spatial Data Based E-Government Applications*. Department of Geodetic and Photogrammetric Engineering, 2004.
- [3] CASTEJÓN GARRIDO, Juan Salvador. *Arquitectura y diseño de sistemas web modernos*. Revista de Ingeniería Informática del CIIRM, 2004.
- [4] EVANS Huw, ASHWORTH Paul. *Getting Started with WAP y WML*. Chapter 1: Introducing WAP and WML, 2006.
- [5] GAONA, Mauricio. FLORIÁN Beatríz. Proceso de Ingeniería Web. Aplicaciones Web e Ingeniería Web. 2007. p. 16-17, 23-25.
- [6] GAONA, Mauricio. Formulación de sistemas y aplicaciones Web. Tendencias en Ingeniería de Software. 2006. p. 9.
- [7] GEO CONSORTIUM Inc. Introduction to Spatial Data Management with Postgis. Disponible en: http://www.mapbender.org/presentations/Spatial_Data_Management_Arnulf_Christl/Spatial_Data_Management_Arnulf_Christl.pdf. [Consultado en Mayo 2008]
- [8] GONZALEZ, Victor. Mapserver y su aplicación a SIG. Universidad técnica particular de Loja, 2005. Disponible en http://sig.utpl.edu.ec/sigutpl/biblioteca/manuales/curso_mapserver.PDF. [Consultado en: Marzo 2008]
- [9] GOSLING James, JOY Bill, GUY Steele, y BRACHA Gilad, *The Java language specification*, Tercera edición. Addison-Wesley, 2005.
- [10] GÜTING, Ralf Hartmut. An Introduction to Spatial Database Systems. Informatikzentrum Universitätsstr. Disponible en: <http://dna.fernuni-hagen.de/papers/IntroSpatialDBMS.pdf>. [Consultado en Abril 2008]
- [11] KNYZIAK Tomasz, WINIECKI Wieslaw. *The new prospects of distributed measurement systems using Javak 2 Micro Edition mobile phone*. Science Direct, 2005.
- [12] MANSOURIAN A, RAJABIFARD A, VALADAN ZOEJ M.J, WILLIAMSON I. Using SDI and web-based system to facilitate disaster management. Science Direct, 2005.

- [13] MARTIN MARTIN Manuel. Manual Postgis. Refrations Inc. Disponible en <http://postgis.refrations.net/docs/postgis.pdf>. [Consultado en: Febrero 2008]
- [14] MINNESOTA UNIVERSIDAD. API de Mapserver. Disponible en <http://mapserver.gis.umn.edu/docs/reference/mapscript>. [Consultado en Mayo 2008]
- [15] ORTIZ, Enrique. Introduction to J2ME Web Services. Sun Developer Network. Disponible en: <http://developers.sun.com/mobility/apis/articles/wsa/>. [Consultado en: Abril 2008]
- [16] PERALTA, Manuel. Sistema de Información. [Fecha de consulta: Mayo11/2007].
- [17] TAN NOZAWA Jaime M. PHP mapscript de Mapserver – Parte 1: Conceptos. 2007. Disponible en <http://phpexperto.blogspot.com/2007/11/php-mapscript-de-mapserver-parte-1.html>. [Consultado en : Abril 2008]
- [18] SABJA RUSS Cristian, SOTO LAGUNAS Alejandro y ANTILLANCA Héctor. Método para transformar el diseño de una aplicación basada en la arquitectura MVC++ a una aplicación Web. Universidad Santiago de Chile, Departamento de Ingeniería Informática. Disponible en: <http://www.diinf.usach.cl/webdiinf/ArchivosSubidos%5C212007184919Articulo%20MVC++%20a%20WEB%20v7.pdf>. [Consultado en Marzo 2008]