



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES

INGENIERÍA EN SISTEMAS

T E S I S

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS.

DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG) PARA LA UBICACIÓN Y CLASIFICACIÓN DE LOS DESASTRES NATURALES DE LA PROVINCIA DE LOJA UTILIZANDO SOFTWARE LIBRE PARA SU DESARROLLO.

AUTORES:

- ◆ LIMA LÓPEZ BYRON RENÉ.
- ◆ HERRERA COFRE DENNYS FABIÁN.

DIRECTORA:

ING. LORENA ELIZABETH CONDE ZHINGRE



LOJA - ECUADOR

2009

DECLARACIÓN DE AUTORÍA:

Nosotros, Byron René Lima López y Dennys Fabián Herrera Cofre, autores de este trabajo de tesis, certificamos la propiedad intelectual a favor de la Universidad Nacional de Loja, los cuales podrán hacer uso del mismo con la finalidad académica que estimen conveniente.

Autores:

.....
Byron René Lima López.
Cofre.

.....
Dennys Fabián Herrera

PENSAMIENTOS:

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el interesante mundo del saber”.

Albert Einstein.



“El tiempo fue inventado por el Todopoderoso para dar una oportunidad a las ideas”.

N.M.B.

“Un libro abierto... es un cerebro que habla; cerrado... un amigo que espera; olvidado... un alma que perdona; destruido... un corazón que llora”.

Proverbio Hindú.

“Las semillas necesitan tiempo para germinar, crecer y producir frutos. Empieza desde ahora a hacer proyectos y a cultivar ideales, y a su tiempo recogerás muchos frutos”.

Guilbert.

“El fruto del silencio es la oración. El fruto de la oración es la fe. El fruto de la fe es el amor. El fruto del amor es el servicio. El fruto del servicio es la paz”.

Madre Teresa de Calcuta

AGRADECIMIENTOS:



Dennys

Herrera

A Dios por haberme dado la fuerza y fortaleza para culminar con éxito mis estudios. A mi familia, por haberme guiado y apoyado para forjarme como profesional de bien. A la Ing. Lorena Conde, directora de tesis, por haber colaborado en la realización de este proyecto de tesis. A la Universidad Nacional de Loja por la formación académica integral que me ha brindado como estudiante.

Byron

Lima.

Agradezco primeramente al Creador, por ser quien guía mi vida en todo momento.

A la universidad Nacional de Loja, paradigma del desarrollo educacional, al Área de Energía, Industrias y Recursos Naturales no Renovables y en particular a la carrera de Sistemas, que abrió las puertas hacia el conocimiento y la formación profesional



con un alto nivel académico. Y de manera muy especial a la Ing. Lorena Conde. Por su dedicación y esmero, supo guiar y dirigir este trabajo.

DEDICATORIA:

Dennys

Herrera

Dedico el presente trabajo de tesis a toda mi familia, y de manera especial a mis queridos padres Laudencio y Elsa, que gracias a su esfuerzo, paciencia, dedicación y apoyo incondicional, logré terminar mis estudios superiores.

A la Universidad Nacional de Loja que me permitió adquirir conocimientos para elevar la cultura general, tanto académica como profesional, para lograr alcanzar grandes metas que me formen como buen profesional para



servir a la sociedad.

Byron

Lima

Dedico este esfuerzo:

A Dios por ser el amigo que nunca falla, por darme la fe y fortaleza necesaria para enfrentar la vida.

A mis abnegado padres Carlos y María, por su gran esfuerzo, sacrificio y ejemplo de trabajo y honradez.

A mi familia y amigos que ha sabido

compartir su tiempo y experiencias.

ÍNDICE GENERAL

Portada	I
Certificación del Director	II
Declaración de Autoría	III
Pensamientos	IV
Agradecimiento	V
Dedicatoria	VI
Índice General	VII



Índice de Figuras.....	XI
Índice de Cuadros.....	XIV
1. RESUMEN	1
2. ABSTRACT	3
3. INTRODUCCIÓN	5
4. METODOLOGÍA	9
4.1 Materiales	10
4.1.1 Propuesta	10
4.1.2 Materiales Necesarios	10
4.1.3 Digitalización de Archivos Shapes (capas)	11
4.2 Métodos	11
4.2.1 Metodología para la Investigación	11
4.2.2 Metodología para el desarrollo del proyecto	12
5. FUNDAMENTACIÓN TEÓRICA.....	18
5.1 Descripción de la Metodología.....	19
5.1.1 Definición de Metodologías.....	19
5.1.1.1 Proceso unificado racional (RUP).....	19
5.1.1.2 Extreme Programming.....	20
5.1.1.3 Método de Desarrollo de Sistema Dinámico (DSDM).....	22
5.1.1.4 Scrum.....	22
5.1.1.5 Iconix.....	23
5.1.2 Diferencias entre Metodologías.....	25
5.2 Sistema de Información Geográfica.....	28
5.2.1 Introducción.....	28
5.2.2 Concepto.....	28
5.2.3 Origen y evolución de los SIGs.....	28
5.2.3.1 Bases Conceptual y Cartográfico de los SIGs.....	28
5.2.3.2 Historia de la Tecnología SIG.....	29
5.2.3.3 Evolución de los SIG.....	31



5.2.3.4	Futuro de los SIG.....	31
5.2.4	Funciones de un SIG.....	35
5.2.5	Aplicaciones de Sobremesa.....	35
5.2.6	Ejemplos de Aplicaciones SIG libres.....	36
5.2.6.1	GRASS GIS.....	36
5.2.6.2	JUMP.....	37
5.2.6.3	GvSIG.....	39
5.2.7	Componentes de los SIGs.....	43
5.2.8	Modelado de Datos Espaciales.....	43
5.2.8.1	Modelos Vectoriales.....	44
5.2.8.2	Modelos de Celdas o Raster.....	44
5.2.8.3	Modelos de superficies en 3 dimensiones.....	45
5.2.9	Shapefile.....	45
5.2.9.1	Tipos de Shapes o Geometrías.....	46
5.3	Software Libre.....	47
5.3.1	Introducción.....	47
5.3.2	Historia.....	49
5.3.3	Libertades de Software Libre.....	50
5.3.4	Tipos de Licencias.....	52
5.3.4.1	Licencia GPL.....	52
5.3.4.2	Licencias estilo BSD.....	54
5.3.4.3	Licencias estilo MPL y derivadas.....	55
5.3.4.4	Copyleft.....	56
5.3.5	Comparación con el software Open Source.....	57
5.4	Desastres Naturales.....	58
5.4.1	Definición.....	59
5.4.2	Clasificación.....	59
5.4.2.1	Desastres Naturales.....	59
5.4.2.1.1	Desastres Generados por Procesos Dinámicos en el Interior de la Tierra.....	59
5.4.2.1.2	Desastres Generados por Procesos Dinámicos en la superficie de la Tierra.....	60



5.4.2.1.3	Desastres Generados por Fenómenos Meteorológicos o Hidrológicos.....	60
5.4.2.1.4	Desastres de Origen Biológico.....	61
5.4.2.2	Desastres Tecnológicos.....	61
5.4.3	Causas que agravan los Desastres.....	61
5.5	Herramientas del Sistema.....	62
5.5.1	Lenguaje de Programación JAVA.....	62
5.5.1.1	Historia.....	62
5.5.1.2	Filosofía.....	63
5.5.1.2.1	Orientado a Objetos.....	63
5.5.1.2.2	Independencia de La Plataforma.....	64
5.5.1.2.3	El Recolector de Basura.....	66
5.5.1.3	Entornos de Funcionamiento.....	67
5.5.1.3.1	En dispositivos móviles y sistemas empotrados.....	67
5.5.1.3.2	En el navegador web.....	68
5.5.1.3.3	En sistemas de servidor.....	68
5.5.1.3.4	En aplicaciones de escritorio.....	69
5.5.1.4	Plataformas soportadas.....	70
5.5.1.5	Industria relacionada.....	70
5.5.1.6	Versiones del lenguaje JAVA.....	71
5.5.2	NetBeans.....	74
5.5.2.1	Historia.....	74
5.5.2.2	NetBeans IDE.....	75
5.5.3	Gestor de Base de Datos MySQL.....	76
5.5.3.1	Historia del Proyecto.....	76
5.5.3.2	Lenguajes de Programación.....	77
5.5.3.3	Aplicaciones.....	77
5.5.3.4	Especificaciones.....	78
5.5.3.5	Plataformas.....	79
5.5.4	Geotools.....	80
5.5.4.1	Introducción.....	80
5.5.4.2	Geotools1 (GT1).....	80
5.5.4.3	Geotools2 (GT2).....	81



5.5.4.4	Geotools2 para la gestión de mapas.....	82
5.5.4.5	Diseño de Geotools2.....	83
5.5.4.6	Instalación de Geotools2.....	69
5.5.4.7	Comparación con otras Librerías.....	85
6.	EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN.....	87
7.	DEASARROLLO DE LA PROPUESTA ALTERNATIVA.....	90
7.1	Definición del Problema.....	91
7.1.1	Pasos a seguir para la conversión de Coordenadas.....	94
7.1.1.1	Sobre la Geometría del Elipsoide.....	95
7.1.1.2	Sobre la latitud y la Longitud.....	96
7.1.1.3	Sobre el Huso.....	97
7.1.1.4	Ecuaciones de Coticchia-Surace para el Problema Directo (Paso de Geográficas a UTM).....	100
7.2	Documentación de Requerimientos.....	102
7.2.1	Requerimientos Funcionales.....	102
7.2.2	Requerimientos no Funcionales.....	106
7.3	Glosario de Términos.....	108
7.4	Modelado.....	111
7.4.1	Modelo de Dominio.....	111
7.4.2	Modelo de Casos de Uso.....	112
7.4.2.1	Caso de Uso # 1.....	113
7.4.2.2	Caso de Uso # 2.....	117
7.4.2.3	Caso de Uso # 3.....	121
7.4.2.4	Caso de Uso # 4.....	128
7.4.2.5	Caso de Uso # 5.....	135
7.4.2.6	Caso de Uso # 6.....	140
7.4.2.7	Caso de Uso # 7.....	146
7.4.2.8	Caso de Uso # 8.....	150
7.4.2.9	Caso de Uso # 9.....	154
7.4.3	Prototipo de Pantallas.....	158
7.4.4	Diagrama de Paquetes.....	178



ÍNDICE DE FIGURAS

FIGURA 1: Representación de un actor	14	181
FIGURA 2: Diagrama Entidad-Relación	14	182
FIGURA 3: Representación de una relación	15	183
FIGURA 4: Tipos de Pruebas Aplicadas al Sistema	17	183
FIGURA 5: Etapas del modelo	19	183
FIGURA 6: Pruebas de Funcionamiento	19	184
FIGURA 7: Pruebas de Aceptación	20	184
FIGURA 8: Modelo de Extreme Programming	22	185
FIGURA 9: Esquema de Trabajo	23	189
FIGURA 10: Esquema de Trabajo de la Metodología	23	189
FIGURA 11: Análisis del Resultado de Pruebas	28	191
FIGURA 12: Componentes de los VALORACION TÉCNICO-ECONÓMICA	31	193
FIGURA 13: Fase de Evolución de los SIG	32	196
FIGURA 14: Recomendaciones	33	198
FIGURA 15: Bibliografía	36	200
FIGURA 16: Interfaz de GRASS	38	
FIGURA 17: Interfaz de SIG Móvil		
FIGURA 18: Interfaz de SIG Móvil		



FIGURA 15: Interfaz de svSIG	39
FIGURA 16: Mapa Conceptual del SW libre	47
FIGURA 17: Geotools2 en Contexto	82
FIGURA 18: Modulo de Geotools2	83
FIGURA 19: Arquitectura del SIG DNPI	93
FIGURA 20: Esquema de Conversión de Coordenadas	94
FIGURA 21: Representación del meridiano central del huso o zona	98
FIGURA 22: Modelo de Dominio	111
FIGURA 23: Casos de Uso	112
FIGURA 24: LIC01	113
FIGURA 25: Diagrama de Robustez # 1	115
FIGURA 26: Diagrama de Secuencia # 1	116
FIGURA 27: LIC02	117
FIGURA 28: Diagrama de Robustez # 2	119
FIGURA 29: Diagrama de Secuencia # 2	120
FIGURA 30: LIC03	121
FIGURA 31: Diagrama de Robustez # 3	124
FIGURA 32: Diagrama de Robustez # 3 (Curso Alterno de Eventos)	125
FIGURA 33: Diagrama de Secuencia # 3	126
FIGURA 34: Diagrama de Secuencia # 3 (Curso Alterno de Eventos)	127
FIGURA 35: LIC04	128
FIGURA 36: Diagrama de Robustez # 4	131
FIGURA 37: Diagrama de Secuencia # 4	132
FIGURA 38: Diagrama de Secuencia # 4 (Curso Alterno de Eventos)	133
FIGURA 39: LIC05	134
FIGURA 40: Diagrama de Robustez # 5	137
FIGURA 41: Diagrama de Secuencia # 5	138
FIGURA 42: Diagrama de Secuencia # 5 (Curso Alterno de Eventos)	139



FIGURA 43: IIC06	140
FIGURA 44: Diagrama de Robustez # 6	143
FIGURA 45: Diagrama de Secuencia # 6	144
FIGURA 46: Diagrama de Secuencia # 6 (Curso Alterno de Eventos)	145
FIGURA 47: IIC07	146
FIGURA 48: Diagrama de Robustez # 7	148
FIGURA 49: Diagrama de Secuencia # 7	149
FIGURA 50: IIC08	150
FIGURA 51: Diagrama de Robustez # 8	152
FIGURA 52: Diagrama de Secuencia # 8	153
FIGURA 53: IIC09	154
FIGURA 54: Diagrama de Robustez # 9	156
FIGURA 55: Diagrama de Secuencia # 9	157
FIGURA 56: Plantilla Instalación de la Base de Datos	158
FIGURA 57: Plantilla Ingreso al Sistema	159
FIGURA 58: Plantilla Pantalla Principal	160
FIGURA 59: Plantilla Información General Provincia	161
FIGURA 60: Plantilla Ayuda SIG DNPI	162
FIGURA 61: Plantilla Archivo Shane	162
FIGURA 62: Plantilla Ventana Desastre Natural	163
FIGURA 63: Plantilla Modificar Desastre Natural	163
FIGURA 64: Plantilla Paleta de Colores	164
FIGURA 65: Plantilla Agregar Nuevo Desastre	165
FIGURA 66: Plantilla Coordenadas Geográficas	165
FIGURA 67: Plantilla Insertar Informe Desastre	166
FIGURA 68: Plantilla Buscar Imágenes	166
FIGURA 69: Plantilla Buscar, Actualizar y Eliminar Desastre	167
FIGURA 70: Plantilla Filtrar Información	168



FIGURA 71: Plantilla Reporte Desastre	168
FIGURA 72: Plantilla Visualizar puntos filtrados	169
FIGURA 73: Plantilla Desastre Natural Generado	170
FIGURA 74: Plantilla Zoom Imagen	171
FIGURA 75: Plantilla Visualizar Informe Desastre	171
FIGURA 76: Plantilla Galería de Imágenes	172
FIGURA 77: Plantilla Reporte Desastre Generado	173
FIGURA 78: Plantilla Reporte Informe General del Desastre	173
FIGURA 79: Plantilla Reporte Imagen	174
FIGURA 80: Plantilla Leyenda Canas	174
FIGURA 81: Plantilla Resaltar Cantón	175
FIGURA 82: Plantilla Resaltar Desastre Generado	176
FIGURA 83: Plantilla Herramientas para manipular canas	176
FIGURA 84: Plantilla Datos Filtrados de Ultima Búsqueda	177
FIGURA 85: Paquete SIG DNPI	178
FIGURA 86: Paquete SIG DNPI negocio	178
FIGURA 87: Paquete SIG DNPI modelo	179
FIGURA 88: Paquete SIG DNPI vistas	180
FIGURA 89: Diagrama de Clases	181
FIGURA 90: Diagrama Entidad Relación	182

ÍNDICE DE CUADROS

CUADRO 1: Comparación de Metodologías por características del proyecto	25
CUADRO 2: Comparación de Metodologías por la curva de Aprendizaje.....	26
CUADRO 3: Referencias de las Diferentes Metodología.....	26-27
CUADRO 4: Principales SIG existentes en el mercado.....	41
CUADRO 5: Tipos de Geometrías.....	46



CUADRO 6: Libertades de SW libre.....	50
CUADRO 7: Funcionalidades del SIG frente a las librerías.....	85
CUADRO 8: Requerimientos del Usuario.....	92
CUADRO 9: Requerimientos Funcionales.....	102-105
CUADRO 10: Requerimientos No Funcionales.....	106-107
CUADRO 11: Caso de Uso # 1.....	113-114
CUADRO 12: Caso de Uso # 2.....	117-118
CUADRO 13: Caso de Uso # 3.....	121-123
CUADRO 14: Caso de Uso # 4.....	128-131
CUADRO 15: Caso de Uso # 5.....	134-136
CUADRO 16: Caso de Uso # 6.....	140-142
CUADRO 17: Caso de Uso # 7.....	146-147
CUADRO 18: Caso de Uso # 8.....	150-151
CUADRO 19: Caso de Uso # 9.....	154-155
CUADRO 20: Resultados de Pruebas.....	186-187
CUADRO 21: Cuadro de Porcentajes.....	188
CUADRO 22: Recursos Humanos.....	190
CUADRO 23: Recursos Técnicos.....	191
CUADRO 24: Recursos Materiales.....	192





1. RESUMEN

RESUMEN:

Los sistemas de Información Geográfica sirven de apoyo para dar solución rápida a diversos problemas que se pueden presentar en el manejo de datos geográficos. Si bien este tipo de aplicaciones son muy necesarias, también es cierto que en muchos casos son demasiado costosas, debido a que demandan mucho esfuerzo y en muchos casos se requiere comprar licencias, que a veces no están al alcance de pequeñas empresas e instituciones que hoy en día necesitan de este tipo de aplicaciones.



El presente trabajo de tesis de grado se basa en el desarrollo de una aplicación SIG para la ubicación y clasificación de Desastres Naturales en la Provincia de Loja utilizando *software libre*, una aplicación de escritorio que ayudará a minimizar el trabajo, acelerar procesos y tomar decisiones acertadas, a los directivos de instituciones vinculadas al entorno geológico, y específicamente al CINFA (Centro Integrado de Geomática Ambiental), ya que para el desarrollo del presente proyecto se contó con el apoyo y colaboración de dicha institución.

Los procesos que se manejan en esta aplicación son: visualización completa de la información para cada desastre natural generado en la provincia de Loja (damnificados, población afectada, imágenes, etc.), ingreso de nuevos desastres ocurridos, transformación de coordenadas de tipo geográficas a planas, actualización de la información existente (desastres, mapas, imágenes), consultas generales y específicas de desastres naturales ocurridos, reportes imprimibles para cada desastre, interacción directa (acercamientos, alejamientos, giros, desplazamientos) con los mapas.

Esta aplicación está desarrollada en base a la Interfaz gráfica de Usuario (GUI) NetBeans para acceder fácilmente a las herramientas, que puede implementarse con diferentes lenguajes de programación pero nosotros utilizamos el lenguaje JAVA; además un sistema manejador de base de datos MySQL para almacenar datos necesarios para el manejo e información de desastres naturales generados.



2. ABSTRACT

ABSTRACT:

The systems of Geographical Information serve as support to give quick solution to diverse problems that can show up in the handling of geographical data. Although this type of applications is very necessary, it is also certain that in many cases they are too expensive, because they demand a lot of effort and in many cases it is required to buy licenses that are not sometimes within reach of small companies and institutions that nowadays need of this type of applications.



The present work of grade thesis is based on the development of an application SIG for the location and classification of Natural Disasters in the County of Loja using *free software*, a desk application that will help to minimize the work, to accelerate processes and to make guessed right decisions, to the directive of institutions linked to the geologic environment, and specifically to CINFA (Center Integrated of Environmental Geomática), since for the development of the present project it counted with the support and collaboration of this institution.

The processes that are managed in this application are: complete visualization of the information for each natural disaster generated in the county of Loja (damaged, affected population, images, etc.), I enter of happened new disasters, transformation of coordinated of geographical type to flat, upgrade of the existent information (disasters, maps, images), you consult general and specific of natural happened disasters, report printable for each disaster, direct interaction (approaches, estrangements, turns, displacements) with the maps.

This application is developed based on User's Graphic Interface (GUI) NetBeans to consent easily to the tools that it can be implemented with different programming languages but we use the language JAVA; also a system database manejador MySQL to store necessary data for the handling and information of natural generated disasters.



3. INTRODUCCIÓN

INTRODUCCIÓN:

La Universidad Nacional de Loja, es una Institución de Educación Superior; que desde el año 1943 viene realizando investigación científico-técnica sobre los problemas del entorno, con calidad, pertinencia y equidad, a fin de coadyuvar al desarrollo sustentable de la región y del país, interactuando con la comunidad, generando propuestas alternativas a los problemas nacionales, con responsabilidad social; reconociendo y promoviendo la diversidad cultural y étnica y la sabiduría popular, apoyándose en el



avance científico y tecnológico, en procura de mejorar la calidad de vida del pueblo ecuatoriano.

La estructura orgánica institucional que actualmente mantiene la Universidad está conformada por las distintas Áreas Académico Administrativas, entre las que consta el Área de Energía, Industrias y Recursos Naturales No Renovables, unidad académica que en su estructura actual mantiene las carreras de Ingeniería en Sistemas, Ingeniería en Electromecánica, Ingeniería en Geología Ambiental y Ordenamiento Territorial, Ingeniería en Electrónica y Telecomunicaciones, Tecnología en Electricidad, Tecnología en Electrónica y las Carreras Artesanales de Técnicas Constructivas, Técnicas Eléctricas, Mecánica Automotriz y Técnicas en Audio y Video.

Siendo la UNL una institución académica de gran prestigio a nivel nacional, el AEIRNNR como parte de la misma, y específicamente la Carrera de Ingeniería en Sistemas adquiere un gran compromiso con la sociedad al formar profesionales con suficientes conocimientos para resolver problemas de la vida cotidiana mediante el desarrollo de sistemas computacionales.

Además en la estructura de la Universidad consta el Área Agropecuaria y de Recursos Naturales Renovables, donde se encuentra el CINFA (Centro Integrado de Geomática Ambiental), que tiene como finalidad el manejo de datos geográficos utilizando tecnología computarizada.

Mediante la colaboración y apoyo de esta institución logramos elaborar el proyecto denominado *SIG para la ubicación y clasificación de Desastres Naturales de la Provincia de Loja*, que tiene como finalidad ofrecer al usuario una interfaz amigable para el ingreso, manipulación y eliminación de desastres naturales que se presenten en la Provincia de Loja; el mismo que nos ha servido para enriquecer un poco más los conocimientos adquiridos en la Universidad, sobre todo en lo que a programación se refiere, por ejemplo utilizando nuevas librerías como son: geotools para el manejo y manipulación de mapas, y JasperReport para la presentación de reportes.



Esta aplicación es de mucha importancia ya que toda la información recolectada referente a desastres naturales en la actualidad solo se almacena en Word o en hojas de Excel, siendo necesario un SIG especializado que permita visualizar esta información en un mapa e ingresar nueva información de manera fácil y eficiente.

Para el desarrollo de la aplicación se plantearon los siguientes objetivos:

- ❖ **Crear el Sistema de Información Geográfica (SIG) para la ubicación y clasificación de los desastres naturales de la Provincia de Loja en un entorno de SW libre.**
- ❖ Sistematizar, Integrar y Organizar la información aislada sobre los diferentes desastres naturales que ocurren en la Provincia de Loja.
- ❖ Diseñar y Construir una Base de Datos (MySQL) que maneje la información relevante (tipo de desastre, ubicación, fecha, coordenadas, imágenes, etc.)
- ❖ Proveer la información precisa de desastres más frecuentes que se presentan en lugares específicos de la Provincia de Loja.
- ❖ Elaborar interfaces para el manejo, migración y estructuración de la información almacenada con miras al acoplamiento de la misma con el SIG y a la interacción con el usuario.
- ❖ Diseñar y Construir el SIG utilizando tecnología JAVA, en base a la información obtenida sobre los desastres naturales que ocurren en la Provincia de Loja.
- ❖ Implementar el SIG en el Centro Integrado de Geomática Ambiental (CINFA) de la UNL.

Los objetivos se cumplen uno a uno durante el desarrollo del proyecto de tesis, y para ello se lo ha dividido en las siguientes partes:



- ✓ *Revisión literaria*, que comprende toda la información correspondiente a Sistemas de Información Geográficos, las metodologías utilizadas para el desarrollo de proyectos de Sistemas, conceptos sobre desastres naturales, herramientas utilizadas (lenguaje de programación, plataforma, base de datos, librerías implementadas).
- ✓ *Materiales*, aquí se describe los materiales que se utilizaron para el desarrollo de la aplicación, como son: sistema operativo, inventario de desastres naturales, librerías, plataforma, lenguaje de programación, software de diseño, base de datos, pluggins
- ✓ *Resultados Obtenidos*, donde se describen todos los requerimientos del usuario (funcionales y no funcionales) y el modelado del sistema mediante los distintos diagramas (modelo de dominio, casos de uso, robustez, secuencia, entidad-relación, clases, paquetes)
- ✓ *Plan de validación*, donde se muestran los resultados obtenidos al haber implementado el sistema, mediante el uso de encuestas a los encargados de realizar el plan, para luego realizar cálculos estadísticos
- ✓ *Conclusiones y recomendaciones*, mismas que se obtuvieron luego del desarrollo del proyecto

El SIG para la Ubicación y Clasificación de Desastres Naturales en la Provincia de Loja, se puede implementar en cualquier institución relacionada con el manejo de desastres naturales en la Provincia, ya que permite obtener e ingresar información completa y clasificada de los desastres generados, ayudando de esta manera a los directivos a tomar decisiones acertadas para evitar futuros problemas o elaborar planes de contingencia de acuerdo a la información obtenida.



4. METODOLOGÍA

4.1 Materiales

A continuación se especificarán los materiales que se utilizaron para el desarrollo de la aplicación SIG para la ubicación y clasificación de Desastres Naturales generados en la Provincia de Loja



4.1.1 Propuesta

Con el propósito de encontrar alternativas de solución ante los desastres naturales que afectan a la Provincia de Loja, lo que proponemos mediante nuestro sistema es:

- ✓ Correcta administración de la información.
- ✓ Rapidez en el acceso a la información
- ✓ Interfaz amigable.
- ✓ Interacción entre los mapas geográficos y la información almacenada.
- ✓ Consultas de una manera interactiva.

4.1.2 Materiales necesarios

Para el desarrollo de la aplicación fueron necesarios los siguientes materiales:

- ✓ El Inventario de todos los desastres naturales ocurridos ahora y en años anteriores, recogido por estudiantes de la carrera de Ingeniería Agronómica y por la Defensa Civil, misma que fue revisada por profesionales especialistas pertenecientes al departamento del CINFA.
- ✓ Librerías Geotools2 para la administración de mapas y JasperReport para la creación de reportes.
- ✓ Computadoras con Sistema Operativo Windows Xp, con lenguaje de programación JAVA, plataforma NetBeans 6.1, software de diseño Enterprise Architect 3.60, base de datos MySql 5.0 y el plugin de IReport para generar los reportes.
- ✓ Procesamiento y corrección de la información cartográfica, misma que fue proporcionada por el personal técnico del CINFA y que se realizó mediante el programa ArcView Gis 8.1.

4.1.3 Digitalización de archivos shapes (capas)



En lo que se refiere a la información cartográfica, como los mapas de la provincia de Loja que son la base de todo el sistema, su digitalización se llevó a cabo tomando en cuenta los siguientes parámetros:

- ✓ La información cartográfica está almacenada en capas que se dividen de acuerdo al contenido que el usuario quiera visualizar.
- ✓ Cada capa contiene información georeferencial que se almacena en archivos con extensión .shp
- ✓ Cada capa consta de sus respectivos atributos, puntos, líneas y polígonos que se utilizan de acuerdo a la información que representan.
- ✓ La capa principal que se utiliza para el manejo de desastres generados sirve para la interacción directa del usuario con el sistema cuando este así lo requiera.

4.2 Métodos

4.2.1 Metodología para la Investigación

Se utilizan los siguientes métodos:

Método Cualitativo Permite determinar la importancia que tiene el uso de un Sistema de Información Geográfica dentro de una organización al momento de tomar decisiones de forma eficiente y responsable, Además muestra características únicas en el manejo correcto de la información recolectada. “De acuerdo con Marcos (1998) este método emplea un conjunto de técnicas que buscan describir, codificar y traducir el significado, no la frecuencia, de ciertos fenómenos que ocurren en el mundo social”¹.

Método Cuantitativo mismo que permitirá determinar la frecuencia con que ocurren los desastres, su magnitud y ubicación exacta, que cantidad de información maneja el sistema y el volumen de datos que se administra.

¹ María del Socorro Marcos (1998). – Manual para la elaboración de Tesis”, Editorial Trillas, México, D.F., 1998.



Método Inductivo, que permitirá integrar toda la información de los desastres naturales ocurridos en la Provincias de Loja para que sean vinculada al Sistema de Información Geográfica mediante una Base de Datos Eficiente.

Método Descriptivo Permite realizar un análisis de cómo funcionan las operaciones de manejo de información de desastres actualmente, orientado a la elaboración de etapas como: descripción del problema, definición y formulación de objetivos, marco teórico, la selección de técnicas y en la descripción, análisis e interpretación de los datos.

4.2.2 Metodología para el Desarrollo del Proyecto

El método escogido para el desarrollo del proyecto es el ICONIX ya que es relativamente ágil y lo suficientemente robusto para un proyecto de mediana envergadura.

Concretamente, se cumplen las siguientes recomendaciones para emplear ICONIX en un proyecto:

- ✓ Interés sincero por todas las partes en que el proyecto tenga éxito.
- ✓ El equipo de trabajo es mediano.
- ✓ A diferencia de XP, con este proceso existe un contrato fijo previo especificando tiempo, recursos y alcance, debido a que se hace uso de una gestión de costes y tiempo.
- ✓ El equipo dispone de una formación elevada y capacidad de aprender.
- ✓ El proyecto tiene un riesgo medio en cuanto a lo innovador de la tecnología.

La metodología de ICONIX tiene éxito dentro de proyectos Open Source. Es relativamente simple, realiza ciertos procedimientos propios de XP como Integración continua, refactorización, pruebas unitarias y Test-Driven Development, a la vez toma las tareas de análisis y diseño de RUP.

El proceso ICONIX contempla 5 fases las cuales comprenden varias actividades, que se deben ejecutar para asegurar el éxito del proyecto.

Las fases que cuenta con el modelo ICONIX son:



1. Análisis
2. Diseño
3. Codificación o Construcción
4. Pruebas
5. Implementación

A continuación se procederá a explicar la aplicación de cada una ellas en este proyecto.

➤ **Análisis.**

Durante la etapa del análisis se recopiló la información necesaria para tener una idea clara del alcance que debe tener el proyecto, que es lo que se debe hacer y cómo se debe hacerlo, así como la estructuración de datos reales necesarios para su desarrollo.

- ✓ *Definición del problema*, permite referirse a los conceptos relacionados con el problema que el sistema está tratando de resolver. El modelo del dominio es la tarea que permitió descubrir "los objetos " (las clases) estos nos representaran posteriormente los conceptos esenciales del sistema.
- ✓ *Requerimientos*, se declararon las partes necesarias del producto final y con ellos se elaboró el sistema de acuerdo a las características específicas de los usuarios, relacionados con sus necesidades.
- ✓ *Glosario de Términos*, nos permitió describir el contenido de los elementos de información es decir todos los datos mencionados en el sistema con una especificación del proceso.
- ✓ *Diagrama de Casos de Uso*, este modelo muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Entre sus componentes están los siguientes:

- **Actores:** Un actor es cualquier entidad, como una persona u otro sistema y que realiza algún tipo de interacción con el sistema.

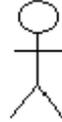


FIGURA 1: Representación de un actor

- **Casos de uso:** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

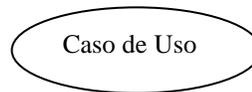


FIGURA 2: Representación de un caso de uso.

- **Relaciones:** Indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

Las relaciones pueden ser de distintos tipos, entre ellas encontramos las siguientes:

<incluye> : Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base. Se suele utilizar para encapsular un comportamiento parcial común a varios casos de uso.

<extiende> : Cuando un caso de uso base tiene ciertos puntos (puntos de extensión) en los cuales, dependiendo de ciertos criterios, se va a realizar una interacción adicional. El caso de uso que extiende describe un comportamiento opcional del sistema (a diferencia de la relación incluye que se da siempre que se realiza la interacción descrita).



FIGURA 3: Representación gráfica de una relación.



- ✓ *Especificación de Casos de Uso*, en donde se especifican las acciones del sistema en cuanto a la interactividad con los usuarios, en este documento se estableció en detalle cómo actuará el sistema en diferentes situaciones.

➤ **Diseño.**

En este paso, se realizan el modelo de dominio, que no es otra cosa sino casos de uso, diagrama de clases, diagrama de paquetes, diagramas de robustez, diagramas de secuencia, diagrama de entidad relación y modelo de pantallas.

- a. *Diagrama de Clases*, se representaron los objetos de la vida real en el plano informático, resultándonos de especial utilidad para las siguientes etapas del desarrollo del sistema.
- b. *Diagrama de Paquetes*, se representaron todos los objetos en forma agrupada de acuerdo a su relación entre objetos.
- c. *Diagrama de Robustez*, une el análisis al diseño asegurando que su texto de caso de uso es correcto. Dirigiendo caminos necesarios de acción y permitiendo continuar descubriendo los objetos.
- d. *Diagrama de Secuencia*, en donde se especifican gráficamente los objetos del sistema. La interacción diseñada nos permitirá detallar la conducta de los objetos y encontrar las clases apropiadas para los atributos y su funcionamiento.
- e. *Modelo Entidad – Relación*, en el cual se ordenan las entidades del sistema, y los datos necesarios para su funcionamiento completo, este modelo se desarrolla de acuerdo con el diagrama de clases para conseguir uno de los objetivos importantes de la metodología orientada a objetos como es la encapsulación.
- f. *Modelo de Pantallas*, es un modelo de interfaz de usuario que poco a poco se irá puliendo con el transcurso del desarrollo del sistema, debido a que es interactivo, reiterativo e incremental.

➤ **Codificación o Construcción**

Durante esta etapa se programan todas las partes del sistema, en base a los requerimientos de las fases anteriores, aquí se incluye las respectivas conexiones a la



base de datos que interactué con el sistema, de igual manera se define el modelo de comunicaciones de cada componente para obtener un solo elemento.

➤ **Pruebas**

Es necesario comprobar la valides del sistema para establecer si cumple o no con los objetivos planteados, si satisface los requerimientos del usuario, para lo cual se debe desarrollar un plan de validación que logre determinar si el sistema cumple con las necesidades para el cual fue creado.

A continuación se especifican los tipos de pruebas que se aplicaron:

- ✓ *Funcionalidad*, Permite asegurar que las especificaciones dadas por el usuario se cumplan correctamente.

- ✓ *Usabilidad*, se valoran parámetros como:
 - Confiabilidad del sistema
 - La utilidad del Sistema
 - La robustez del sistema (Interfaz).

➤ **Implementación**

La etapa final del modelo ICONIX es la implementación, que tiene como fin la instalación del SW ya terminado, además se debe cumplir con la entrega de manuales del usuario y programador para el manejo adecuado del sistema.

La Figura 4 ilustra todas las etapas del modelo Iconix adaptadas a nuestro proyecto, que se acopla perfectamente a todo el desarrollo del mismo:





FIGURA 4: Etapas del modelo Iconix



5. FUNDAMENTACIÓN TEÓRICA

5.1 Descripción de la Metodología

Para que el proyecto cumpla con los requerimientos necesarios existe la necesidad de utilizar diversas metodologías que permitan definir las diferentes fases que se van a desarrollar a lo largo del mismo, es decir, para garantizar que cumpla las exigencias para la aplicación y verificación los procedimientos de desarrollo.

5.1.1 Definición de metodologías

Entre las metodologías para el desarrollo de SW que se encuentran disponibles, se definirán las que cumplen las condiciones necesarias para el desarrollo del proyecto.

5.1.1.1 Rational Unified Process (RUP)

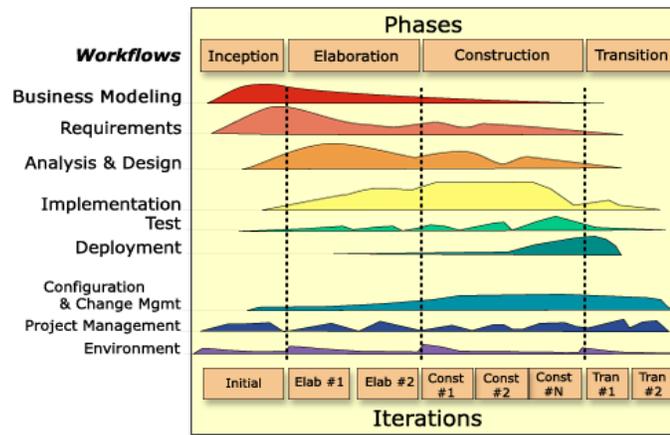


FIGURA 5: Proceso Unificado Rational

RUP es un proceso formal que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. (Customización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

Fases:

Las cuatro fases del ciclo de vida son:

- Concepción
- Elaboración
- Construcción
- Transición



Ventajas:

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Es sencillo ya que sigue pasos intuitivos necesarios a la hora de desarrollar sw.
- Seguimiento detallado en cada una de las fases.

Desventajas:

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Estamos poniendo a nuestro cliente en una situación que puede ser muy incómoda para él.
- Nuestro cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

5.1.1.2 Extreme Programming (XP)

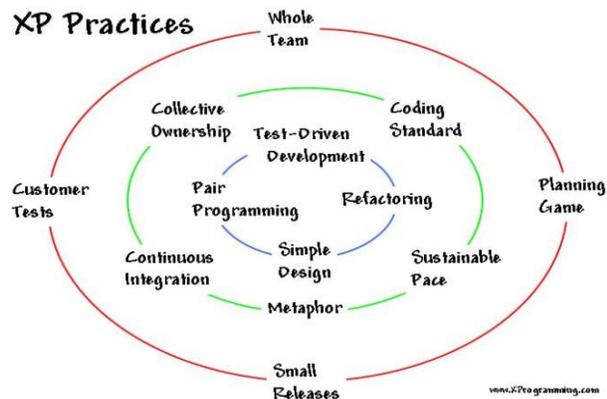


FIGURA 6: Modelo de Extreme Programming²

Es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen

² www.XProgramming.com



que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales del método son:

- Desarrollo iterativo e incremental, es decir pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión
- Programación por parejas
- Frecuente interacción del equipo de programación con el cliente o usuario
- Corrección de todos los errores antes de añadir nueva funcionalidad
- Refactorización del código
- Propiedad del código compartida
- Simplicidad en el código

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Ventajas

- Apropiado para entornos volátiles
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para nuestros clientes, conocen las fechas de entrega de funcionalidades. Vital para su negocio
- Permitirá definir en cada iteración cuales son los objetivos de la siguiente
- La presión esta a lo largo de todo el proyecto y no en una entrega final

Desventajas

- Delimitar el alcance del proyecto con nuestro cliente

Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia.

5.1.1.3 Método de Desarrollo de Sistema Dinámico (DSDM)

Se desarrolló para completar lo que le faltaba al método RAD (método rápido de desarrollo) al proporcionar una estructura que tome en cuenta el ciclo de desarrollo completo. Las características principales del método DSDM son las siguientes:

- Participación del usuario.
- Desarrollo iterativo y creciente.
- Frecuencia de entrega mejorada.
- Pruebas integradas en cada fase.
- La aceptación de los productos entregados depende directamente del cumplimiento de los requisitos.

5.1.1.4 Scrum

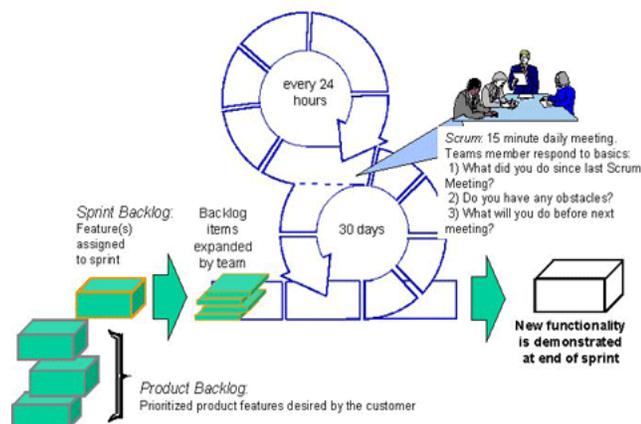


FIGURA 7: Esquema de trabajo SCRUM

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas.



Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto a las necesidades del cliente.

En Scrum, el equipo se focaliza en una única cosa: construir software de calidad. Además tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa.

5.1.1.5 Iconix

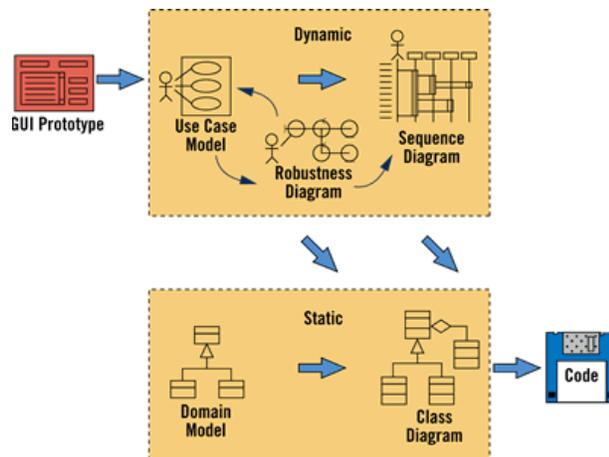


FIGURA 8: Esquema de trabajo de la metodología ICONIX

El proceso de ICONIX es un modelado de objetos conducido por casos de uso, como RUP; también es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño de éste. Este proceso también hace uso aerodinámico del UML mientras guarda un enfoque afilado en el seguimiento de requisitos. Y, el proceso se queda igual a la visión original de Jacobson del “manejo de casos de uso”, esto produce un resultado



concreto, específico y casos de uso fácilmente entendibles, que un equipo de un proyecto puede usar para conducir el esfuerzo hacia un desarrollo real. El enfoque es flexible y abierto; siempre se puede seleccionar de los otros aspectos del UML para complementar los materiales básicos. Además de las características ya mencionadas, presenta las siguientes:

- Centrado en datos: se descompone en fronteras de datos
- Basado en escenarios que descomponen los casos de uso
- Enfoque iterativo e incremental
- Ofrece trazabilidad

Estas técnicas se aplican a proyectos con un equipo de tamaño mediano. Las tres características fundamentales de Iconix son:

1.- Iterativo e incremental.- Varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos (casos de uso, análisis de robustez y el diagrama de secuencia).

2.- Trazabilidad.- Cada paso está referenciado por un requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos

3.- Dinámica del UML.- La metodología ofrece un uso “dinámico del UML” como los diagramas de Caso de Uso, diagramas de Secuencia, de Colaboración

Concretamente, se cumplen las siguientes recomendaciones para emplear ICONIX en un proyecto:

- Interés sincero por todas las partes en que el proyecto tenga éxito.
- El equipo de trabajo es mediano.
- A diferencia de XP, con este proceso existe un contrato fijo previo especificando tiempo, recursos y alcance, debido a que se hace uso de una gestión de costes y tiempo.
- El equipo dispone de una formación elevada y capacidad de aprender.
- El proyecto tiene un riesgo medio en cuanto a lo innovador de la tecnología.



En conclusión la metodología de ICONIX tiene éxito dentro de proyectos desarrollados bajo software libre. Es relativamente simple, realiza ciertos procedimientos propios de XP como Integración continua, refactorización, pruebas unitarias y Test-Driven Development, a la vez toma las tareas de análisis y diseño de RUP, es decir es un proceso intermedio entre XP y RUP, siendo el primero muy útil para software's pequeños y, el segundo, muy útil para software's industriales; por tanto, ICONIX es una mezcla entre la agilidad de XP y la robustez de RUP.

5.1.2 Diferencias entre Metodologías

A continuación presentamos un cuadro de comparación por cada aspecto analizado y lo ponemos a consideración:

Modelo de Proceso	Tamaño del Proyecto	Tamaño del Equipo	Complejidad del Problema
RUP	Medio / Extenso	Medio / Extenso	Medio / Alto
ICONIX	Pequeño / Medio	Pequeño / Medio	Pequeño / Medio
XP	Pequeño / Medio	Pequeño	Medio / Alto
SCRUM	Pequeño / Medio	Pequeño	Medio / Alto

CUADRO 1: Comparación de Metodologías por las características del proyecto

En este cuadro (Cuadro 1) se presenta una comparativa de los modelos de proceso en cuanto a las características del proyecto, analizamos el tamaño del proceso, del equipo y la complejidad del problema para cada uno de los modelos. Podemos resaltar que: con un pequeño equipo de desarrollo se puede realizar grandes proyectos, de alta complejidad; es el caso de XP y SCRUM.

Con respecto a la curva de aprendizaje (Cuadro 2), vemos que algunos modelos nos ofrecen una mayor ventaja pero con ciertas limitaciones, ya que aun no hay sido explotadas a gran escala como lo es RUP que posee alto soporte y herramientas integrales que nos guían a través del mismo, facilitando aplicar con mayor efectividad esta metodología, permitiendo aprovecharla al máximo.



Modelo de Proceso	Curva de aprendizaje	Herramienta de integración	Soporte Externo
RUP	Lenta	Alto Soporte	Alto Soporte
ICONIX	Rápida	Algún Soporte Disponible	Algún Soporte Disponible
XP	Rápida	No mencionado	Algún Soporte Disponible
SCRUM	Rápida	No mencionado	Algún Soporte Disponible

CUADRO 2: Comparación de Metodologías por la curva de Aprendizaje

El siguiente cuadro (Cuadro 3) muestra una comparación entre las diferentes metodologías pero con respecto a sus ventajas y desventajas, así como las consideraciones que se hacen para dicha metodología y las llaves para el éxito.

	Ventajas	Consideración	Desventajas	Llaves para el éxito
Rup	Mucho soporte. Utilizado y difundido en la industria.	Casos de Uso y arquitectura	Complejidad de customización y adaptación. No hay guías disponibles para estudiantes	Hay que ajustarlo a las necesidades de la organización.
Iconix	Fácil de adaptar e implementar. Conjunto pequeño de práctica.	Existe mucha información disponible.	No considera aspectos de Management.	Proyecto de negocio pequeño/mediano, basado en objetivos y/o comparaciones.
Xp	Prácticas basadas en valores fáciles de transmitir. Buen ambiente de trabajo.	Se puede combinar con otros enfoques ágiles.	Management sin mucho detalle. Problema con proyectos grandes.	Guarda con implementación a medias.
Scrum	Ideal para problemas complejos y ambientes	Metodología liviana y fácil de entender	Prácticas de desarrollo y testing no descriptas.	Compromiso e involucramiento del usuario. Decisión de la



	cambiantes.			organización para utilizarlo.
Dsdm	Involucramiento del usuario. Soporte de un consorcio de empresas	Identificar todos los tipos de usuarios.	Hay que pertenecer al consorcio	Hay que aceptar la filosofía de trabajo. Estabilidad de las persona y los skills.

CUADRO 3: Referencias de las Diferentes Metodología

En conclusión, y al analizar los cuadros comparativos de las metodologías, el método Iconix resulta fácilmente adaptable a proyectos de tamaño mediano como lo es nuestro proyecto, frente a las otras metodologías debido a que es fácil de adaptar e implementar.

5.2 SISTEMA DE INFORMACION GEOGRÁFICA

5.2.1 Introducción



Los sistemas de información geográfica (SIG) son hoy en día una herramienta ya de uso masivo. Existe una gran cantidad de software orientado a



facilitar el manejo de los datos vectoriales sobre representaciones digitales, aunque el costo de estos es bastante elevado.

Por otra parte, también es posible encontrar librerías y aplicaciones orientadas al manejo de **FIGURA 9: Componentes de los SIG**³ datos geográficos de forma gratuita. Cada vez son más los recursos de los cuales se dispone para emprender un desarrollo de estas características. Sobre la plataforma Java se desarrollan librerías libres cada vez mejores, conforme el lenguaje crece. Geotools es uno de estos proyectos.

5.2.2 Concepto

“Un Sistema de Información Geográfica (SIG o GIS, en su acrónimo inglés) es una integración organizada de hardware, software, datos geográficos y personal, diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. 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.”⁴

5.2.3 Origen y Evolución de los Sistemas de Información Geográfica

5.2.3.1 Bases conceptual y cartográfico de los SIG

En la historia de los SIG se deben tener en cuenta los conceptos espaciales que facilitaron el desarrollo de la tecnología SIG, así como la historia de Informática, tecnologías que trabajan con información espacial, ciencias afines y Geografía (Gutiérrez y Gould, 1994).

En la década de los 50 y 60, aparecen una serie de conceptos y métodos que han sido la base sobre la cual se desarrollo la tecnología SIG. Bunge (1961) en su libro “*Theoretical Geography*” los identificó como elementos geométricos fundamentales en la Geografía y son con los que trabajan los software SIG vectoriales actuales. Hagertrand (1952) se interesó por el análisis de las superficies, en sus modelos de difusión de innovaciones basado en la compartimentación del espacio por medio de una retícula, que constituye en un antecedente de los software GIS raster. Otros autores como Ullman

³ <http://www.monografias.com/trabajos/gis/gis.shtml>

⁴ http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica



(1956), Garrinson (1960), Nystuen (1961) y Kansky (1963) introdujeron los elementos topológicos en el análisis espacial, y Berry (1964) al tratamiento estadístico de la información Geográfica.

En el año de 1988 la Fundación de Ciencias de Estado Unidos de América crea el Centro Nacional para la Investigación Geográfica y Análisis (NCGIA) por con el objeto de “desarrollar investigaciones básica sobre análisis geográfico utilizando los Sistemas de Información Geográfica” (Bosque, 1992).

“En Europa, es en el Reino Unido donde aparecen algunas de las iniciativas más interesantes, en forma casi paralela en el tiempo a los desarrollados en EE.UU., y los organismos más destacados son Ordnance Survey (OS) y Experimental Cartographic Unit (ECU) (Comas y Ruíz, 1993).”⁵

5.2.3.2 Historia de la Tecnología SIG

SIG en Canadá (Años sesenta)

El Sistema de Información Geográfica de Canadá (CGIS), se considera como el sistema pionero, inició su creación en 1964, financiado por el Departamento de Agricultura de Canadá y en su desarrollo jugó un papel determinante Roger Tomlinson (BOSQUE SENDRA 1992). El CGIS se diseñó para una aplicación específica con el fin de llevar a cabo análisis con la base de datos (tipo inventario). Desde el punto de vista conceptual el CGIS aportó nuevas ideas que están vigentes en los software SIG actuales como por ejemplo: La estructuración de información en capas temáticas, división de mapas digitales en hojas y ajuste en los bordes, topología de arcos, superposiciones topológicas, entre otras (Gutiérrez y Gould, 1994).

Sistemas desarrollados por la Universidad de Harvard (Entre 1966 y 1980).

“El *Laboratory for computer graphics and Spatial Analysis* (Universidad de Harvard USA), es una de las instituciones que más ha aportado en el tema de modelo de datos. Entre los desarrollos de Software se puede clasificar en tres etapas (Bosque, 1992):

⁵http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/int_origenyevo_luciondelossig.htm



- ✓ **Etapa de 1966-1970.** Desarrollo de software de cartografía asistida por computador SYMAP (*SYnagraphic MAPing*) en 1968. También se elaboraron en este periodo se desarrollaron otros sistemas raster como GRID, IMGRID y posteriormente el *Analysis Package* (MAP), en los cuales se desarrollaron las funcionalidades de Análisis raster basados en el Concepto de Superposición de capas (Gutiérrez y Gould, 1994).
- ✓ **Etapa de 1970-1980.** Se desarrolló el modelo de datos basado en una nueva estructura que incorporaba relaciones topológicas como la contigüidad y la conectividad, sobre un programa de conversión de datos (al añadir la topología arco-nodo a los datos “spaguetti” de los CAD), denominado POLYVRT. En este período también se desarrollo la estructura de datos denominado DIME (Dual Independent Map Encoding), el cual fue uno de los primeros que incluía explícitamente la topología de la información espacial. (Bosque, 1992 ; Gutiérrez y Gould, 1994).
- ✓ **Etapa Posterior a 1980.** A partir de la experiencia de POLYVRT, el Laboratorio de Harvard, desarrolló ODYSSEY un software SIG de tipo vectorial. Este incluye al digitalización semiautomática, la gestión de la base de datos, y la elaboración interactiva de los mapas”.⁶

5.2.3.3 Evolución de los SIG

Se pueden distinguir tres fases en la evolución de los SIG (Crain y MacDonald, 1984; Gutiérrez y Gould, 1994):

- **Fase de Inventario.** Esta fase se caracteriza por las aplicaciones relacionadas con grandes inventarios de datos, como las inventarios de redes públicas, transporte, o el catastro.(Localización, condición)

⁶http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/int_origenyevo luciondelossig.htm

- **Fase de Análisis.** En esta fase los SIG son capaces de resolver preguntas más complejas, que exigen relacionar distintas capas de información y utilizar técnicas estadísticas y de análisis espacial. Ejemplo la localización de rellenos sanitarios, ubicación de centros de atención, supermercados, etc.(Condición, tendencias)
- **Fase de Gestión.** Está orientada hacia la gestión y la decisión. Se hace énfasis en análisis espacial sofisticado y en la modelización. (Rutas, pautas y modelos).

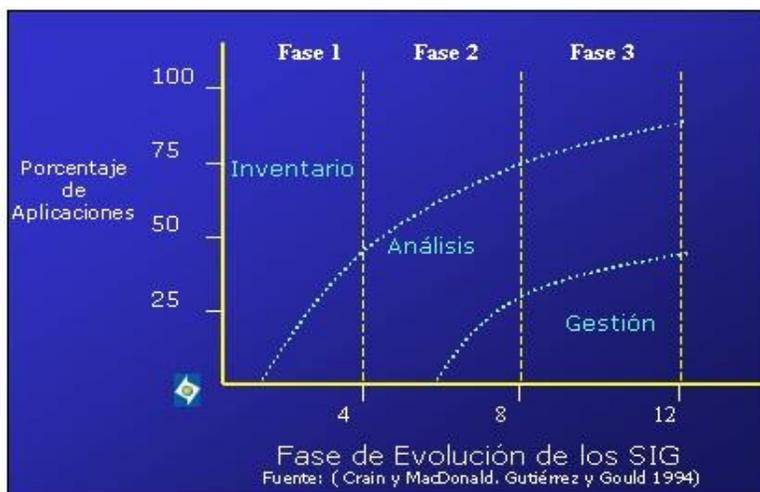


FIGURA 10: Fase de Evolución de los SIG

5.2.3.4 Futuro de los Sistemas de Información Geográfica

La tecnología Internet permitirá difundir el conocimiento geográfico alrededor del mundo. Los SIG ayudarán a la planificación, diseño y construcción de ciudades, obras civiles y organización y protección del ambiente en que vivimos (IGAC-CIAF. Notas de Clase Fundamentos de SIG, 2004)

En la actualidad y en el futuro inmediato los sistemas de información geográfica jugarán un papel muy importante en la implementación de temas como:

✓ **Agricultura de Precisión**

Con el poder analítico del SIG, ahora es posible integrar métodos de colección de datos de campo con otras fuentes de datos, tales como la percepción remota y los sistemas de posicionamiento global (GPS), para proporcionar una herramienta poderosa



que permita maximizar los rendimientos de los cultivos, al redistribuir la cantidad de fertilizantes teniendo en cuenta el potencial productivo de cada parcela; minimizar la aplicación de insumos, maximizar las ventajas financieras y, reducir los impactos en el medio ambiente, en razón de que el tratamiento homogéneo de los campos implica, casi siempre, una sobredosis de insumos que no son aprovechados por las plantas, derivando en la contaminación del agua, del suelo y del aire (León, 2002).

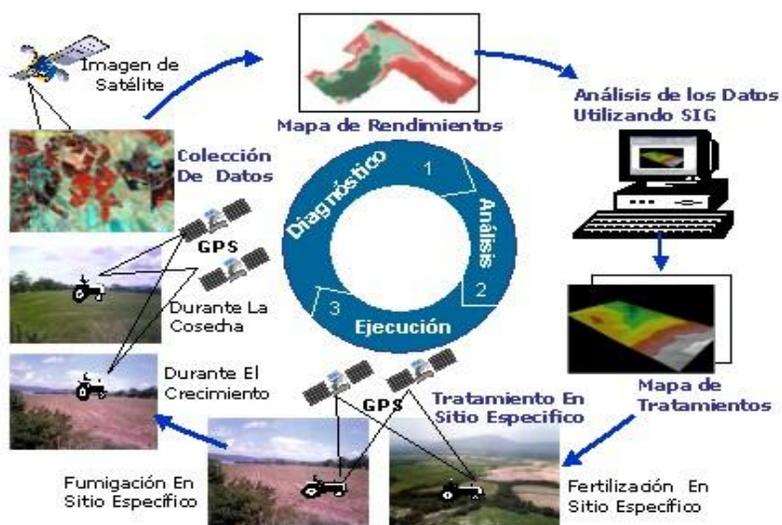


FIGURA 11: Sistema de Posicionamiento Global

✓ **SIG Móvil (GIS movil)**

SIG Móvil es la unificación de tres tecnologías diferenciadas encaminadas a la gestión eficiente de los objetos puntuales y espaciales allí donde se requiere. Cualquier objeto que necesita ser conservado supone un gasto y por lo tanto es necesario acercarse al mismo para inventariarlo o hacer un seguimiento.



FIGURA 12: GIS móvil⁷

✓ **Servicios Web de Información Geográfica**

Un Servicio WEB se puede definir como un “componente de aplicación programable al que se puede acceder mediante protocolos estándar de Internet, lo cual va a garantizar las comunicaciones de extremo a extremo” (Sabando Grasa, 2003), es decir entre el Cliente y el Proveedor del Servicio WEB a través de un protocolo de estándar de comunicaciones en Internet (HTTP).

En el campo concreto de los SIG, se espera a corto plazo adicionar a las funcionalidades actuales como “servidores de mapas” (Map Server) en la Web, servicios de acceso a información territorial abiertos e interoperables en Internet. En este ámbito el Open GIS Consortium OGC (<http://www.opengis.org/>) ha desarrollado una importante labor de estandarización enfocada hacia la interoperación de los sistemas SIG en Web (Sabando Grasa, 2003).

✓ **Ventajas de los Servicios WEB para los SIG**

La explotación centralizada de una información geográfica distribuida es la principal ventaja, siendo esta una gran oportunidad en la superación de las fronteras territoriales de las administraciones públicas competentes en la información geográfica.

Reducción de la duplicidad de fuentes de información incoherentes sobre una misma realidad. Los servicios de localización (por ejemplo de dispositivos móviles) pueden gracias a esta tecnología ampliar sus ámbitos de acción tendiendo hacia la universalidad (Sabando Grasa, 2003).

✓ **Navegación en 3 dimensiones**

Otra de las tendencias de futuro en los Sistemas de Información Geográfica en Internet es la posibilidad del paso de las funcionalidades de los SIG en 2D a 3 dimensiones (3D).

⁷ <http://www.rtm.es>



La información necesaria para la aplicación de un sistema de este tipo es el modelo digital del terreno, y la superposición sobre el mismo de la ortofotografía o imagen de satélite georreferenciada es la mejor opción visual.

Es evidente las mejoras que aporta la tercera dimensión en la navegación por el territorio, por su mayor representatividad y también muy importante por las posibilidades de análisis visual que ofrece. (Sabando Grasa, 2003)

✓ **Infraestructura de Datos Espaciales (IDE -SDI)**

En la actualidad se realizan esfuerzos en todo el mundo para mejorar la coordinación e intercambio de datos entre las entidades encargadas de producir información geográfica a nivel nacional e internacional, como una necesidad de ordenamiento y unificación de los datos espaciales distribuidos en las diferentes entidades públicas y privadas, de manera que su interacción y acceso sea de manera rápida y eficiente.

"El creciente interés en la información geográfica para el desarrollo sostenible, ha llevado a muchos países y organizaciones a adoptar una combinación de técnicas, políticas y mecanismos organizacionales encaminados a compartir la información espacial a través de sus grupos de trabajo. Estos mecanismos son conocidos como Infraestructuras de Datos Espaciales (IDE). Las IDE incentivan la capacidad de los países, los gobiernos locales y las organizaciones para compartir conocimientos e información espacial entre los diversos grupos de trabajo. Las IDE han sido adoptadas a través del mundo entero, en diferentes grados (Borrero, 2002) y América Latina y El Caribe no han sido la excepción, con muchos países haciendo esfuerzos para poner en funcionamiento al menos algunos de los elementos de las IDE." (Rey Martínez, 2003)⁸

5.2.4 Funciones de un SIG.

Las principales cuestiones que puede resolver un Sistema de Información Geográfica, ordenadas de menor a mayor complejidad, son:

⁸http://www.igac.gov.co:8080/igac_web/UserFiles/File/ciaf/TutorialSIG_2005_26_02/paginas/int_origenyevo_luciondelossig.htm



- ✓ **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.

“Por ser tan versátiles los Sistemas de Información Geográfica, su campo de aplicación es muy amplio, pudiendo utilizarse en la mayoría de las actividades con un componente espacial. La profunda revolución que han provocado las nuevas tecnologías ha incidido de manera decisiva en su evolución”.⁹

5.2.5 Aplicaciones SIG de sobremesa.

Se puede considerar Sistemas de Información Geográfica de sobremesa como aquellas aplicaciones SIG orientadas al trabajo normal de visualización, análisis, edición y salidas gráficas de información geográfica. Por otro lado estarían los servidores de datos geográficos como Mapserver , Geoserver o las Geodatabases como PostGIS y MySQL Spatial, que no están cubiertos aquí ya que entrarían en otro ámbito más extenso.

A la hora de presentar en esta ponencia las alternativas de software libre a los SIG comerciales se ha procurado hacer una selección de los sistemas más difundidos y con más funcionalidades existentes actualmente.

5.2.6 Ejemplos de Aplicaciones SIG en SW libre.

5.2.6.1 GRASS GIS

⁹ http://es.wikipedia.org/wiki/Sistema_de_Informaci3n_Geogr3fica

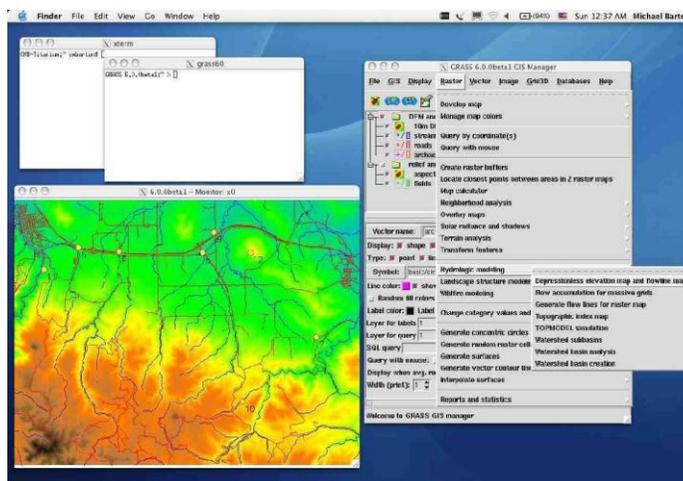


FIGURA 13: Interfaz de GRASS GIS¹⁰

GRASS (*Geographic Resources Analysis Support System*) fue inicialmente concebido y desarrollado por el laboratorio de investigación del cuerpo de ingenieros del ejército de los Estados Unidos (USA-CERL) para la gestión del territorio y la gestión medioambiental.

GRASS comenzó a difundirse en ámbitos educativos y de instituciones públicas y se desarrollaron numerosas aplicaciones alrededor de dicho sistema, hasta que en 1.999 pasó a tener licencia del tipo GNU GPL. Hasta entonces se distribuía de manera más o menos gratuita pero previa autorización del cuerpo de ingenieros del ejército americano. Fue entonces cuando el desarrollo ya no dependía de una sola institución, sino que miles de desarrollares de todo el mundo podían contribuir al futuro de desarrollo de nuevas versiones con nuevas funcionalidades.

Al ser GRASS uno de los SIG con más tiempo de rodaje, el número de herramientas y utilidades que presenta es muy elevado. Originalmente GRASS estaba muy orientado al aspecto matricial (raster) de la información, aunque contaba con un potente editor de topología vectorial, sin embargo en las últimas versiones el aspecto vectorial y sobre todo la conexión a bases de datos externas se ha ido potenciando. En cuanto a la interfaz de usuario también ha experimentado una gran evolución y si bien antes todo el control se hacía por medio de comandos tipo UNIX, en la actualidad y sobre todo con la última versión (6.0) se han producido mejoras considerables en la interfaz del usuario.

¹⁰ Página oficial de GRASS <http://grass.itc.it/>



Uno de los inconvenientes principales de GRASS ha sido precisamente que está diseñado para entornos UNIX, lo que le ha dado gran difusión en centros universitarios y de investigación pero que ha frenado su expansión hacia el público general. Actualmente existen versiones de GRASS que se pueden instalar en entornos Windows a través de emulación de Cygwin.

Puntos fuertes

- ✓ Solidez por los orígenes militares y la edad del proyecto.
- ✓ Herramientas de análisis raster y potente modelado hidrológico.
- ✓ Editor de topología.

Puntos débiles

- ✓ Interfaz no muy amigable.
- ✓ Diseñado para entornos UNIX/Linux
- ✓ Complejidad de uso.

5.2.6.2 JUMP

Jump fue uno de los primeros SIG gratuitos y por lo tanto ha servido de base a otros desarrollos, tanto públicos como desarrollado por empresas de programación. Su origen está en Canadá, ya que nace como un proyecto patrocinado por una serie de instituciones canadienses.

Se trata de un SIG modular programado en Java y que basa su funcionalidad en módulos (*plugins*). De esta forma si queremos cargar cualquier tipo de imagen o dato vectorial sólo tenemos que encontrar o programar el módulo necesario. Lo mismo ocurre con cualquier funcionalidad adicional que se desee implementar: consultas, ediciones avanzadas, etc.

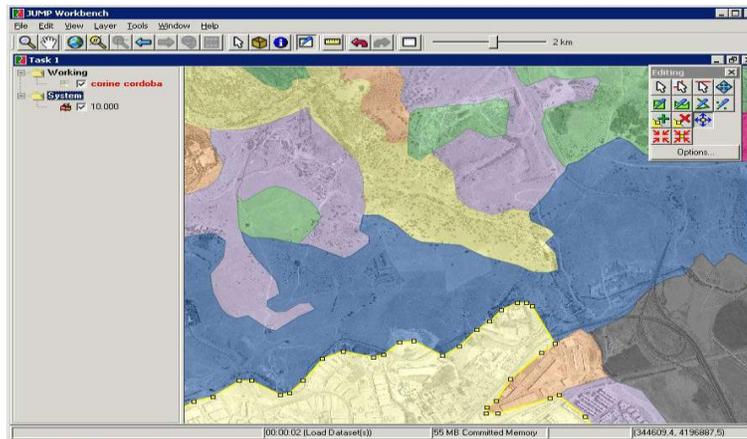


FIGURA 14: Interfaz de JUMP¹¹

La interfaz de usuario es similar a la que proporciona ArcView, con una tabla de contenidos a la izquierda y una ventana central para el mapa. Es posible conectarse a servidores de cartografía WMS y existen plugins para numerosos de formatos tanto de archivo como de servidores. Una de las cosas más interesantes son las herramientas de edición de que se disponen para modificar datos vectoriales, así como herramientas básicas de geoprocado (zonas de influencia, uniones, etc).

Actualmente han aparecido versiones internacionalizadas y varias páginas que albergan proyectos relativos a Jump, tanto para la creación de nuevos plugins como proyectos que basándose en Jump pretenden generar nuevos programas con funcionalidades más específicas.

Puntos fuertes

- ✓ Interfaz de usuario muy intuitiva
- ✓ Buen número de formatos soportados a través de plugins, incluyendo conexión a servidores.
- ✓ Buen punto de partida para la creación de proyectos personalizados debido a la documentación existente y a la facilidad de implementación de nuevas funciones.

Puntos débiles

- ✓ Se echan en falta algunas funcionalidades básicas como por ejemplo la impresión de cartografía, cuadrículas, etc. Muchas están en vías de solución.

¹¹ Página oficial de Jump. <http://www.jump-project.org/>



- ✓ Cierta descoordinación en la generación de versiones, aunque actualmente se ha creado un comité para coordinar el desarrollo de las futuras versiones.

5.2.6.3 GvSIG

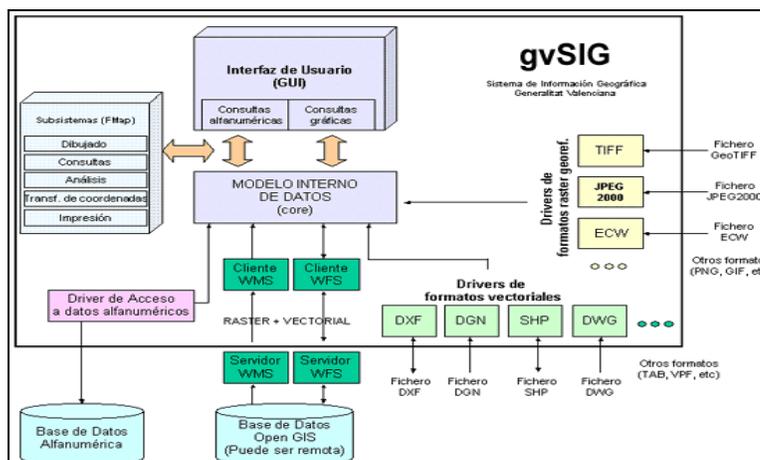


FIGURA 15: Interfaz de gvSIG¹²

gvSIG surge como un proyecto amparado por la Generalitat Valenciana, quien a finales de 2003 sacó a concurso un proyecto para el desarrollo de un SIG con una serie de características como que fuese multiplataforma, de código abierto, modular (Fig. gvSIG.), sin licencias, interoperable con formatos de otros programas (Autocad, Microstation, Arcview, etc) y por supuesto sujeto a estándares del OGC. El resultado ha sido una aplicación, que aunque actualmente se encuentra en fase de desarrollo, ya están disponibles varias versiones al público y con gran parte de la funcionalidad cubierta.

Aunque en la versión actual no existen herramientas de edición, dichas herramientas están en fase de programación, sin embargo, al contrario de otros SIG libres que sí contemplan estas opciones (p.e. Jump), las funciones básicas que cualquier usuario desearía como diseño de impresión, soporte de formatos de imagen típicos como ECW o MrSID, están incorporadas sin necesidad de ningún plugin adicional.

Puntos fuertes

- ✓ Producto muy terminado y orientado al usuario final, tanto a nivel de interfaz de usuario como de funciones implementadas.

¹² Página oficial de gvSIG. <http://www.gvsig.gva.es/framesesp.htm>



- ✓ Soporte para los formatos más populares tanto vectoriales como de imágenes.
- ✓ Funcionalidades previstas muy completas.
- ✓ Totalmente en español.

Puntos débiles

- ✓ Actualmente no soporta edición de datos vectoriales.
- ✓ No permite enlazar tablas (JOIN).



Software SIG	Windows	Mac OS X	GNU/Linux	BSD	Unix	Entorno Web	Licencia de software
ArcGIS	Sí	No	No	No	Sí	Sí	Software no libre
Autodesk Map	Sí	No	No	No	No	Sí	Software no libre
Caris	Sí	No	No	No	No	Sí	Software no libre
CartaLinux	Sí	No	No	No	No	No	Software no libre
Geomedia	Sí	No	No	No	Sí	Sí	Software no libre
GeoServer	Sí	Sí	Sí	Sí	Sí	Java	Libre: GPL
GRASS	Sí	Sí	Sí	Sí	Sí	Mediante pyWPS	Libre: GPL
gvSIG	Java	Java	Java	Java	Java	No	Libre: GPL
IDRISI	Sí	No	No	No	No	No	Software no libre
ILWIS	Sí	No	No	No	No	No	Libre: GPL
Generic Mapping Tools	Sí	Sí	Sí	Sí	Sí	Sí	Libre: GPL
JUMP	Java	Java	Java	Java	Java	No	Libre: GPL
Kosmo	Java	Java	Java	Java	Java	En desarrollo	Libre: GPL
Manifold	Sí	No	No	No	No	Sí	Software no libre
MapGuide Open Source	Sí	Sí	Sí	Sí	Sí	LAMP/WAMP	Libre: LGPL
MapInfo	Sí	No	Sí	No	Sí	Sí	Software no libre
MapServer	Sí	Sí	Sí	Sí	Sí	LAMP/WAMP	Libre: BSD
Mapitude	Sí	No	No	No	No	Sí	Software no libre
MapWindow GIS	Sí (ActiveX)	No	No	No	No	No	Libre: MPL
MicroStation Geographics	Sí	Abandonado	No	No	Abandonado	Sí	Software no libre
Quantum GIS	Sí	Sí	Sí	Sí	Sí	Sí	Libre: GPL
SAGA GIS	Sí	Sí	Sí	Sí	Sí	No	Libre: GPL
SavGIS	Sí	No	No	ND	No	Integración con Google Maps	Software no libre: Freeware
SPRING	Sí	No	Sí	No	No	No	Software no libre: Freeware
TatukGIS	Sí	No	No	No	No	?	Software no libre
INTMips	Sí	No	No	No	Sí	Sí	Software no libre
TransCAD	Sí	No	No	No	No	Sí	Software no libre

CUADRO 4: Principales SIG existentes en el mercado, sistemas operativos en los que pueden funcionar sin emulación, su tipo de licencia.



El cuadro anterior (Cuadro 4) ilustra los principales SIG que se encuentran disponibles en el mercado, y realiza una comparación de cada uno de ellos, de acuerdo al sistema operativo en el que pueden funcionar y el tipo de licencia que tienen.

Como se puede observar casi todos los SIGs que se encuentran en la lista funcionan bajo el sistema operativo Windows, y algunos de ellos utilizan el lenguaje Java, pero así mismo podemos observar que la mayoría de ellos no están desarrollados bajo software libre, lo que los hace demasiado costosos y prácticamente inalcanzables para usuarios comunes de este tipo de sistemas.

Es por ello que actualmente el desarrollo de SIGs, así como de otro tipo de sistemas está apuntando mucho más hacia la utilización del software libre, básicamente debido al costo de las licencias.

En lo que respecta a nuestro sistema está completamente desarrollado bajo software libre, por lo que no se tienen que pagar licencias y servirá como base para futuras modificaciones del mismo, además es un ejemplo que se puede seguir para el desarrollo de otro tipo de aplicaciones.



5.2.7 Componentes de los SIG

Un SIG no es exclusivamente un programa de computador, sino un conjunto de componentes interrelacionados a saber: equipos, programas, bases de datos y recursos humanos.

- ✓ Equipos: compuestos por el computador y sus diferentes dispositivos periféricos, incluyendo dentro de ellos los que permitan la entrada y salida de la información geográfica en diversos medios y formas.
- ✓ Programas: Son los conjuntos de comandos especializados que actúan sobre la información e incluyen los procedimientos de aplicación diseñados por lo usuarios; así, para ser efectivos estos programas deben proveer una base funcional que sea adaptable y expandible de acuerdo a los requerimientos de la organización.
- ✓ Bases de datos: Son el conjunto de datos espaciales y descriptivos que posee el usuario, los cuales contienen toda la información que garantizará el funcionamiento analítico del SIG.
- ✓ Recursos humanos: Son los grupos de personas debidamente capacitadas y con la concepción del manejo de datos gráficos.

5.2.8 Modelación de datos espaciales

En todo sistema de información geográfica, se utiliza un modelo encargado de representar de manera simplificada determinados aspectos de la realidad. Implícitamente se desecha la información que se considera de poco valor; por esta razón el modelo más apropiado varía según los requerimientos que dependen no solo del tipo de problema a resolver, sino también de las capacidades computacionales con las que se cuenta. Por ejemplo, existen modelos que pueden representar mucho mejor la realidad que otros, pero a cambio requieren una mayor cantidad de recursos (mayor tiempo de cómputo, mayor cantidad de memoria, etc.) Los dos principales modelos con los que se representan las entidades espaciales son los modelos Vectoriales y Raster:



5.2.8.1 Modelos Vectoriales

La principal característica del modelo vectorial es el almacenamiento de coordenadas en dos o tres dimensiones usando un formato de números decimales de alta precisión (en una computadora actual, la mantisa debe ser superior a los ocho dígitos). Estas coordenadas son la base para tipos de datos más complejos como son:

- a) Punto:** El más básico, corresponde a un solo grupo de coordenadas que describe una posición única en el espacio.
- b) Líneas:** Se representan con las coordenadas de los puntos inicial y final que describen un segmento de línea.
- c) Arcos circulares:** Requiere la posición del punto inicial, punto final y el centro de la circunferencia a la que pertenecen.
- d) Circunferencias:** Se compone de las coordenadas de punto central y radio o punto sobre la circunferencia.
- e) Polígonos:** Conformado por las coordenadas de los vértices; los polígonos son un conjunto de segmentos de línea con extremos comunes.
- f) Curvas cuadráticas, cúbicas, etc.:** Simula la correspondiente curva que mejor se ajusta a una serie de puntos.

5.2.8.2 Modelos de celdas o raster

El modelo raster representa los datos de tipo gráficos por medio de una matriz de celdas de igual tamaño, de forma que cada celda contiene un atributo (por ejemplo una altura representada por un color). El número de celdas que conforman la malla en ambas direcciones define la resolución, siendo mejor entra más celdas se tengan. (Obviamente el número de celdas es inversamente proporcional el tamaño de estas). Los datos en formato raster son muy simples y su consumo de memoria es proporcional solo al tamaño de la matriz y no a la diversidad y distribución de los datos dentro de ella, lo que lo hace adecuado para representar características que varían suavemente en el espacio, como ocurren con las fotos digitalizadas, mapas de relieve, etc. Por otro lado, no son aptos para la manipulación pues pierden definición con las rotaciones o cambios de escala.



5.2.8.3 Modelos de superficies en tres dimensiones

Los modelos en tres dimensiones se emplean para describir una superficie en el espacio, donde los puntos que la conforman no solo poseen una localización sobre el plano horizontal si no también una altura respecto al mismo. Son muy utilizados cuando el relieve de una región es de importancia, como ocurre en estudios topográficos, o hidrológicos incluyendo la determinación de zonas con mayor riesgo de inundación, comportamiento de una cuenca, etc. También se pueden utilizar como una manera de representar otros valores que varían espacialmente.

5.2.9 Shapefile

La empresa ESRI propietaria de ArcView (software de administración y visualización geográfica) crea el formato Shapefile, que es un conjunto de archivos que contienen las geometrías y sus datos asociados.

Tres son los tipos de archivos que forman parte de este formato:

- ✓ Shapefile (*.shp): Almacena las geometrías.
- ✓ DBase (*.dbf): Almacena los atributos descriptivos.
- ✓ Index (*.shx): Archivo de índice. La correspondencia entre los registros de un dbf y un shp es de uno es a uno.

El dbf y el shx deben tener el mismo nombre que tiene el shp para ser reconocidos como parte de él.



5.2.9.1 Tipos de Shapes o Geometrías

Los shapes son almacenados en los shapefiles y pueden ser de los siguientes tipos:

Valor	Shape Tipo
0	Shape Nulo
1	Punto
3	Polilínea
5	Polígono
8	Multipunto
11	PuntoZ
13	PoliLineaZ
15	PolígonoZ
18	MultiPuntoZ
21	PuntoM
23	PoliLineaM
25	PolígonoM
28	MultiPuntoM
31	MultiPatch

CUADRO 5. Tipos de Geometrías

Los números que no se encuentran en la lista están reservados para desarrollos futuros. Los shapefiles están limitados a almacenar sólo un tipo de geometría. Un mapa está representado por un gran conjunto de información vectorial, que se organiza por medio de capas (coberturas). Cada capa está formada por geometrías de un mismo tipo, como por ejemplo polígonos, y cada una de estas capas se almacena en un Shapefile.

Además, con motivo de mantener un orden conceptual sobre las geometrías, las capas suelen subdividirse según lo que estas representan, es decir, todas las geometrías que representan por ejemplo Los ríos dentro del mapa, se almacenan en un Shapefile.



5.3 Software Libre

5.3.1 Introducción

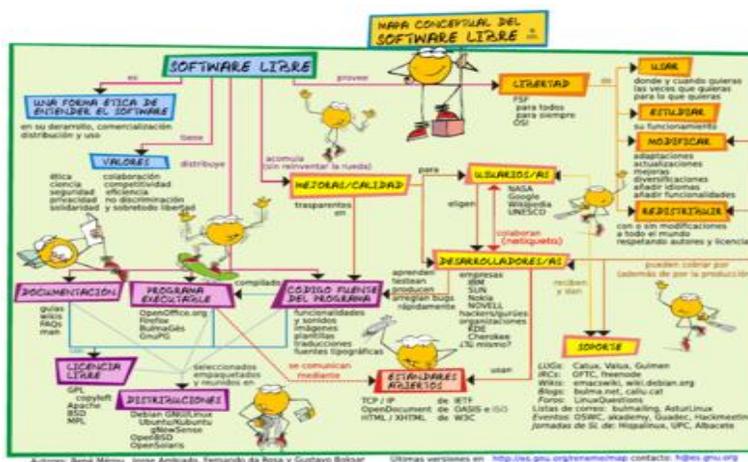


FIGURA 16: Mapa Conceptual del SW libre¹³

Software libre (en inglés **free software**) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo).

El software libre suele estar disponible gratuitamente, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por ende no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en algunas ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo

¹³ http://es.wikipedia.org/wiki/Código_libre



sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquél que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público.

5.3.2 Historia

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época (los mainframes) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

Allá por el 1971, cuando la informática todavía no había sufrido su gran boom, las personas que hacían uso de ella, en ámbitos universitarios y empresariales, creaban y compartían el software sin ningún tipo de restricciones.

Con la llegada de los años 80 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que esta lo solucionara. Aunque el programador estuviese capacitado para solucionar el



problema y lo desease hacer sin pedir nada a cambio, el contrato le impedía que mejorase el software.

“El mismo Richard Stallman cuenta que por aquellos años, en el laboratorio habían recibido una impresora donada por una empresa externa. El dispositivo, era utilizado en red por todos los trabajadores, parecía no funcionar a la perfección dado que cada cierto tiempo el papel se atascaba. Como agravante, no se generaba ningún aviso que se enviase por red e informase a los usuarios de la situación.

La pérdida de tiempo era constante, ya que en ocasiones, los trabajadores enviaban por red sus trabajos a imprimir y al ir a buscarlos se encontraban la impresora atascada y una cola enorme de trabajos pendientes. Richard Stallman decidió arreglar el problema, e implementar el envío de un aviso por red cuando la impresora se bloqueara. Para ello necesitaba tener acceso al código fuente de los controladores de la impresora. Pidió a la empresa propietaria de la impresora lo que necesitaba, comentando, sin pedir nada a cambio, que era lo que pretendía realizar. La empresa se negó a entregarle el código fuente.

En ese preciso instante, Richard Stallman se vio en una encrucijada, debía elegir entre aceptar el nuevo software privativo firmando acuerdos de no revelación y acabar desarrollando más software privativo con licencias restrictivas, que a su vez deberían ser más adelante aceptadas por sus propios colegas.

Con este antecedente, en 1984, Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la Free Software Foundation (FSF). Stallman introdujo una definición para free software y el concepto de "copyleft", el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software".¹⁴

¹⁴ http://es.wikipedia.org/wiki/C3digo_libre#cite_note-1



5.3.3 Libertades del software libre

De acuerdo con la definición anterior, el software es "libre" si garantiza las siguientes libertades:

Libertad 0	Libertad 1	Libertad 2	Libertad 3
Ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, militar, etc.)	Estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente)	Copiar el programa de manera que se pueda ayudar al vecino o a cualquiera	Mejorar el programa y publicar las mejoras
Es importante señalar que las libertades 1 y 3 obligan a que se tenga acceso al código fuente.			
La "libertad 2" hace referencia a la libertad de modificar y redistribuir el software libremente licenciado bajo algún tipo de licencia de software libre que beneficie a la comunidad.			

CUADRO 6: Libertades del SW libre

Ciertos teóricos usan este cuarto punto (libertad 3) para justificar parcialmente las limitaciones impuestas por la licencia GNU GPL frente a otras licencias de software libre (ver Licencias GPL). Sin embargo el sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que podría ser resuelta en la próxima versión 3.0 de la licencia GNU GPL, causa en estos momentos graves perjuicios a la comunidad de programadores de **software libre**, que muchas veces no pueden reutilizar o mezclar códigos de dos licencias distintas, pese a que las libertades teóricamente lo deberían permitir.

En el sitio web oficial de OSI está la lista completa de las licencias de software libre actualmente aprobadas y tenidas como tales.



El término software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del *copyright*; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

“La definición de software libre no contempla el asunto del precio; un eslogan frecuentemente usado es *“libre como en libertad, no como en cerveza gratis”* o en inglés *“Free as in freedom, not as in free beer”* (aludiendo a la ambigüedad del término inglés *“free”*), y es habitual ver a la venta CDs de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo. El software gratis puede incluir restricciones que no se adaptan a la definición de software libre —por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc

Para evitar la confusión, algunas personas utilizan los términos "libre" (*Libre software*) y "gratis" (*Gratis software*) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término *open source software* (software de código abierto, también llamado de fuentes abiertas). La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos.

En un intento por unir los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra "FLOSS" con el significado de "Free - Libre - Open Source Software" e, indirectamente, también a la comunidad que lo produce y apoya".¹⁵

¹⁵[http:// es.wikipedia.org/wiki/C3digo_libre](http://es.wikipedia.org/wiki/C3digo_libre)



5.3.4 Tipos de Licencias

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarlo. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

5.3.4.1 Licencias GPL

La controvertida **L**icencia **P**ública **G**eneral GNU será analizada en primer lugar. Sin dudas junto con la licencia estilo BSD, son las más conocidas en el mundo del software libre. Como todo lo que proviene de la FSF y por añadidura de Richard Stallman, esta licencia no escapa del centro de la polémica.

Algunos se refieren a la GNU GPL como de naturaleza viral. Defienden esta postura indicando que la misma infecta a los programas con el virus de la libertad. Esto es porque un programa que está protegido por la GPL no puede transformarse en software propietario. Esta principal falencia que remarcan los que la critican, es la virtud más importante que defienden los que están a su favor. El concepto que hay detrás de esta licencia es el de copyleft.

Copyleft deriva de un juego de palabras que representan lo contrario de copyright. De hecho, el copyleft incluye la registración de los derechos de autor.

Este concepto fue acuñado en la FSF y se encuentra enmarcado por la GNU GPL. El proceso consiste en reservar los derechos sobre un programa y luego añadirle los términos de distribución (por ejemplo la GPL). Estos términos son el instrumento legal que le dan a todo el mundo los derechos de utilizar, modificar y redistribuir el código fuente del programa o cualquier programa derivado del mismo. Todo esto es posible si los términos de distribución no son cambiados.

En el primer párrafo de la licencia, queda bien clara la intención de la misma, que es heredada directamente de los ideales propugnados por Stallman.



'las licencias para la mayoría de los programas se crean para quitarte tu libertad. En cambio, la GNU GPL pretende garantizar tu libertad de compartir y modificar software libre; Asegurar que el software sea libre para todos los que lo usan'.

Esta licencia se aplica a cualquier programa u otro trabajo que contenga un aviso del titular del derecho de autor que puede distribuirse bajo los términos de la Licencia Pública General GNU.

Actos Permitidos:

- ✓ Distribuir copias de software libre.
- ✓ Modificar software libre y redistribuirlo.
- ✓ Cobrar por el acto de transferir una copia.
- ✓ Ofrecer garantía a cambio de un cánon.
- ✓ No publicar las modificaciones mientras se usen en forma privada. Esto incluye a las empresas mientras mantengan los cambios dentro de su ámbito.

Actos NO Permitidos:

- ✓ Imponer nuevas restricciones a la licencia.
- ✓ Copiar, modificar, sublicenciar o distribuir el programa de una manera distinta de la expresamente utilizada por la licencia.

A primera vista, puede parecer que la GPL no permite la convivencia con un intento comercial relacionado con el software libre. El modelo tradicional de ganar dinero a través de la venta de copias solamente no es posible. Pero la GPL puede ser extraordinariamente efectiva para establecer una plataforma que desaliente la creación de nuevas plataformas competitivas. Se establece un único campo donde todas compiten en el mismo nivel y donde ser el primero tiene muchos beneficios.

Un ejemplo de esto es la empresa Cygnus Solutions. Cygnus generó durante muchos años cambios al compilador gcc, entre ellos portarlo a nuevos tipos de arquitecturas de hardware. La gran mayoría de sus trabajos cumplen con la GPL, y luego se incluían a la distribución de gcc. Cygnus cobra por el esfuerzo involucrado en la portación y mantenimiento a sus clientes, pero no por el código fuente.



“Si una empresa intenta competir a la par de Cygnus, se verá forzada a redistribuir su trabajo. De esta forma se beneficia en primer lugar Cygnus ya que la competencia no puede diferenciarse por la plataforma tecnológica. El cliente elige por el nivel de servicio. Por otro lado también se beneficia toda la comunidad de software libre que recibe las mejoras al compilador tan utilizado”.¹⁶

5.3.4.2 Licencias estilo BSD

Dentro del mundo del software libre, las licencias estilo BSD han sido muy importantes y muy utilizadas. Su origen se remonta a las raíces del movimiento. Esta licencia fue la primera que se ideó para distribuir software libre de las entregas BSD. Estas entregas fueron la forma en que el CSRG distribuía su trabajo alrededor del sistema operativo UNIX.

La primera vez que se utilizó esta licencia fue en la distribución Networking Release 1. En la actualidad, se sigue utilizando como licencia para varios proyectos. Entre los más importantes se encuentran:

- ✓ Los sistemas operativos: FreeBSD, NetBSD y OpenBSD.
- ✓ El servidor web Apache.
- ✓ El sistema de bases de datos PostgreSQL.

Detalles Importantes:

- ✓ La principal diferencia de las licencias estilo BSD y las de la familia de la GPL es que los cambios efectuados pueden publicarse en forma binaria/ejecutable sin distribuir el código fuente.
- ✓ No se entrega ninguna garantía sobre el correcto funcionamiento del software.
- ✓ Redistribuciones del código fuente deben mantener los avisos de derecho de autor, la lista de condiciones y la negación de garantía.

¹⁶ <http://www.gnu.org/copyleft/gpl.html>



5.3.4.3 Licencias estilo MPL y derivadas

La Netscape Public License fue desarrollada por Netscape cuando lanzó como Código Fuente Abierto a su producto Netscape Navigator. Actualmente esta versión del navegador se la conoce como Mozilla.

Varios hackers famosos dentro del movimiento de Código Fuente Abierto, entre ellos Linus Torvalds, Bruce Perens y Eric Raymond colaboraron como consultores ad-honorem durante el desarrollo de la licencia. Aunque intentaron persuadir a Netscape para que utilizase la GPL, su esfuerzo fue en vano. Terminaron lanzándolo bajo la NPL que cumple con la Definición de Código Fuente Abierto.

Fue la primera licencia nueva luego de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD.

Antes de lanzar su código fuente al público, Netscape liberó una versión beta de su licencia, el 5 de marzo de 1998, en un grupo de noticias especialmente creado para opinar sobre la misma (netscape.public.mozilla.license). Esto despertó gran entusiasmo y derivó en propuestas varias para modificar algunos términos de la NPL.

La sección de la licencia que fue más criticada es la que le confiere a Netscape privilegios especiales como es la posibilidad de relicenciar modificaciones hechas por cualquiera al código. También pueden tomar estas modificaciones, mejorarlas y negarse a contribuir las al proyecto.

Esta previsión fue creada porque Netscape tenía contratos con las compañías que proveían módulos que estaban incluidos en el navegador (en total 75 módulos). Este aspecto de la licencia hizo suponer que la misma no sería aceptada finalmente por la comunidad de Código Fuente Abierto.



La gente de Netscape tuvo en cuenta el feedback recibido y para solucionar este problema se creó la Mozilla Public License. Ambas licencias son idénticas, salvo que la NPL mantiene las cláusulas que protegen los derechos de Netscape.

El código fuente de Netscape Navigator fue liberado originalmente bajo la NPL y todas las modificaciones deben lanzarse bajo la misma licencia. Pero si se desarrollan nuevos módulos de código, pueden lanzarse bajo la licencia MPL o alguna compatible (obviamente la GPL no lo es).

Detalles Importantes de la MPL

- ✓ Los cambios deben volver al proyecto.
- ✓ “Cualquier individuo o compañía que contribuye al código del proyecto debe renunciar a cualquier derecho de patentamiento sobre el código fuente”.¹⁷

5.3.4.4 Copyleft

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre (por ejemplo MySQL); esta práctica *no* restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft. También podría retirar todas las licencias de software libre anteriormente otorgadas, pero esto obligaría a una indemnización a los titulares de las licencias en uso. “En España, toda obra derivada está tan protegida como una original, siempre que la obra derivada parta de una autorización contractual con el autor. En el caso genérico de que el autor retire las licencias “copyleft”, no afectaría de ningún modo a los productos derivados anteriores a esa retirada, ya que no tiene efecto retroactivo. En términos legales, el autor no tiene derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional”.¹⁸

¹⁷ <http://www.opensource.org/licenses/mozilla1.1.php>

¹⁸ <http://es.wikipedia.org/wiki/Copyleft>



5.3.5 Comparación con el software Open Source

Aunque en la práctica el software Open Source y el software libre comparten muchas de sus licencias, la FSF opina que el movimiento Open Source es filosóficamente diferente del movimiento del software libre. Apareció en 1998 con un grupo de personas, entre los que cabe destacar a Eric S. Raymond y Bruce Perens, que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto.

Estos defensores ven que el término *open source* evita la ambigüedad del término inglés *free* en *free software*. El término "open source" fue acuñado por Christine Peterson del *think tank* Foresight Institute, y se registró para actuar como marca registrada para los productos de software libre.

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. (Véase también La Catedral y el Bazar). El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento Open Source ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la FSF se distancia tanto del movimiento Open Source como del término "Open Source".

Puesto que la OSI sólo aprueba las licencias que se ajustan a la OSD (*Open Source Definition*), la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "open source" con "software libre". Aun cuando existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal software, raramente suelen tener impacto en el proceso de colaboración.

"Aunque el término "Open Source" elimina la ambigüedad de Libertad frente a Precio (en el caso del Inglés), introduce una nueva: entre los programas que se ajustan a la *Open Source Definition*, que dan a los usuarios la libertad de mejorarlos, y los



programas que simplemente tiene el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier software que tenga el código fuente disponible es *open source*, puesto que lo pueden manipular (un ejemplo de este tipo de software sería el popular paquete de software gratuito Graphviz, inicialmente no libre pero que incluía el código fuente, aunque luego AT&T le cambió la licencia). Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios”.¹⁹

5.4 Desastres Naturales

Debido a que los desastres naturales revierten los avances en materia de desarrollo alcanzados con enorme esfuerzo y engendran pobreza en tantas partes del mundo, hoy en día muchas organizaciones ya sea gubernamentales o no, están orientadas hacia la creación de un futuro más seguro y para ello se plantean ideas novedosas en torno a actividades y proyectos creativos para la gestión de riesgos, ya sea mediante programas de donaciones, o mediante gestión de las propias organizaciones.

Es así que acontecimientos recientes en nuestro país han demostrado la importancia del manejo de desastres en la planificación del desarrollo y la frecuencia con que este tema se pasa por alto. Las graves inundaciones surgidas en el 2008 dejaron consecuencias desastrosas, dejando a comunidades enteras afectadas a nivel nacional y especialmente en la provincia de Loja.

Como todos sabemos los desastres no respetan personas ni lugares. Golpean en cualquier parte: en los países en desarrollo, donde en parte son responsables de mantener a muchas personas en condiciones de pobreza, y en las naciones occidentales industrializadas.

Los desastres pueden infligir reveses importantes al progreso económico y social de los países en desarrollo y con mucha frecuencia causan desgracias masivas que desvían los preciados recursos destinados a objetivos de desarrollo concretos, es por ello que

¹⁹ http://es.wikipedia.org/wiki/Open_Source



existen organizaciones como la Defensa Civil que ayudan a los damnificados y además previenen la aparición de nuevos desastres.

5.4.1 Definición

Los desastres son alteraciones intensas de las personas los bienes, los servicios y el medio ambiente, causadas por un suceso natural o generado por el hombre, que exceden la capacidad de respuesta de la comunidad afectada.

5.4.2 Clasificación

Según el criterio de la Organización de las Naciones Unidas (ONU), los desastres se clasifican de la siguiente manera:

- ✓ Desastres Naturales, y
- ✓ Desastres Tecnológicos.

5.4.2.1 Desastres Naturales

Son los desastres producidos por la fuerza de la naturaleza. Entre estos tenemos:

5.4.2.1.1 Desastres Generados por Procesos Dinámicos en el Interior de la Tierra

a. Sismos.- Son los movimientos de la corteza terrestre que generan deformaciones intensas en las rocas del interior de la tierra, acumulando energía que súbitamente es liberada en forma de ondas que sacuden la superficie terrestre.

b. Tsunamis.- Movimiento de la corteza terrestre en el fondo del océano, formando y propagando olas de gran altura.

c. Erupciones Volcánicas.- Es el paso del material (magma), cenizas y gases del interior de la tierra a la superficie.



5.4.2.1.2 Desastres Generados por Procesos Dinámicos en la Superficie de la Tierra

a. Deslizamiento de Tierras.- Que ocurren como resultado de cambios súbitos o graduales de la composición, estructura, hidrología o vegetación de un terreno en declive o pendiente

b. Derrumbes.- Es la caída de una franja de terreno que pierde su estabilidad o la destrucción de una estructura construida por el hombre

c. Aludes.- Masa de nieve que se desliza pendiente abajo.

d. Aluviones.- Flujos de grandes volúmenes de lodo, agua, hielo, rocas, originados por la ruptura de una laguna o deslizamiento de un nevado.

e. Huaycos.- Desprendimientos de lodo y rocas debido a precipitaciones pluviales, se presenta como un golpe de agua lodosa que se desliza a gran velocidad por quebradas secas y de poco caudal arrastrando piedras y troncos.

5.4.2.1.3 Desastres Generados por Fenómenos Meteorológicos o Hidrológicos

a. Inundaciones.- Invasión lenta o violenta de aguas de río, lagunas o lagos, debido a fuertes precipitaciones fluviales o rupturas de embalses, causando daños considerables. Se pueden presentar en forma lenta o gradual en llanuras y de forma violenta o súbita en regiones montañosas de alta pendiente.

b. Sequías.- Deficiencia de humedad en la atmósfera por precipitaciones pluviales irregulares o insuficientes, inadecuado uso de las aguas subterráneas, depósitos de agua o sistemas de irrigación.

c. Heladas.- Producidas por las bajas temperaturas, causando daño a las plantas y animales.



d. Tormentas.- Fenómenos atmosféricos producidos por descargas eléctricas en la atmósfera.

e. Granizadas.- Precipitación de agua en forma de gotas sólidas de hielo.

f. Tornados.- Vientos huracanados que se producen en forma giratoria a grandes velocidades.

g. Huracanes.- Son vientos que sobrepasan más 24 Km./h como consecuencia de la interacción del aire caliente y húmedo que viene del océano Pacífico con el aire frío.

5.4.2.1.4 Desastres de Origen Biológico

a. Plagas.- Son calamidades producidas en las cosechas por ciertos animales.

b. Epidemias.- Son la generalización de enfermedades infecciosas a un gran número de personas y en un determinado lugar.

5.4.2.2 Desastres Tecnológicos

- a. Incendios.
- b. Explosiones.
- c. Derrames de Sustancias Químicas.
- d. Contaminación Ambiental.
- e. Guerras.
- f. Subversión.
- g. Terrorismo.

5.4.3 Causas que agravan las catástrofes.

- ✓ La posición en una región de alta actividad Tectónica (Terremotos y Vulcanismos).
- ✓ El Clima lluvioso y tempestuoso inestable.
- ✓ Intervención Antrópica fuerte sobre el ambiente.
- ✓ Crecimiento Urbano Inadecuadamente Controlado



- ✓ Flujo Poblacional desde las Zonas Rurales hacia las Urbanas.

5.5 Herramientas del Sistema

Este tema tiene una especial importancia en el desarrollo de la aplicación, ya que se describen las herramientas con las que se trabajó durante todo el proyecto, y solo una correcta elección de las mismas nos permitió un desarrollo óptimo del sistema

5.5.1 Lenguaje de Programación JAVA.

5.5.1.1 Historia

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.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre)



5.5.1.2 Filosofía

El lenguaje Java se creó con cinco objetivos principales:

- ✓ Debería usar la metodología de la programación orientada a objetos.
- ✓ Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- ✓ Debería incluir por defecto soporte para trabajo en red.
- ✓ Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- ✓ Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker Architecture), Internet Communications Engine o OSGi respectivamente.

5.5.1.2.1 Orientado a Objetos

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que use estén unidos a sus operaciones. Así, 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). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas



reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

5.5.1.2.2 Independencia de la Plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es lo que significa ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run everywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode) —instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).



Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características “dependientes” de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acatación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una “recompilación dinámica” en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento



que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o "Escríbelo una vez, ejecútalo en cualquier parte" por "Escríbelo una vez, depúralo en todas partes")

5.5.1.2.3 El Recolector de Basura

Un argumento en contra de lenguajes como C++ es que los programadores se encuentran con la carga añadida de tener que administrar la memoria solicitada dinámicamente de forma manual:

En C++, el desarrollador puede asignar memoria en una zona conocida como *heap* (montículo) para crear cualquier objeto, y posteriormente desalojar el espacio asignado cuando desea borrarlo. Un olvido a la hora de desalojar memoria previamente solicitada puede llevar a una *fuga de memoria*, ya que el sistema operativo seguirá pensando que esa zona de memoria está siendo usada por una aplicación cuando en realidad no es así. Así, un programa mal diseñado podría consumir una cantidad desproporcionada de memoria. Además, si una misma región de memoria es desalojada dos veces el programa puede volverse inestable y llevar a un eventual *cuelgue*. No obstante, se debe señalar que C++ también permite crear objetos en la pila de llamadas de una función o bloque, de forma que se libere la memoria (y se ejecute el destructor del objeto) de forma automática al finalizar la ejecución de la función o bloque.

En Java, este problema potencial es evitado en gran medida por el recolector automático de basura (o *automatic garbage collector*). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el



responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste (que, desde un punto de vista de bajo nivel es una dirección de memoria). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aun así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios—es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y puede que más rápida que en C++.

La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

Debe tenerse en cuenta que la memoria es sólo uno de los muchos recursos que deben ser gestionados.

5.5.1.3 Entornos de Funcionamiento

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática.

5.5.1.3.1 En dispositivos móviles y sistemas empotrados

Desde la creación de la especificación J2ME (Java 2 Platform, Micro Edition), una versión del entorno de ejecución Java reducido y altamente optimizado, especialmente desarrollado para el mercado de dispositivos electrónicos de consumo se ha producido toda una revolución en lo que a la extensión de Java se refiere.



Es posible encontrar microprocesadores específicamente diseñados para ejecutar bytecode Java y software Java para tarjetas inteligentes (JavaCard), teléfonos móviles, buscapersonas, set-top-boxes, sintonizadores de TV y otros pequeños electrodomésticos. El modelo de desarrollo de estas aplicaciones es muy semejante a las *applets* de los navegadores salvo que en este caso se denominan *MIDIlets*.

5.5.1.3.2 En el navegador web

Desde la primera versión de java existe la posibilidad de desarrollar pequeñas aplicaciones (Applets) en Java que luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web. Estas mini-aplicaciones se ejecutan en una JVM que el navegador tiene configurada como extensión (*plug-in*) en un contexto de seguridad restringido configurable para impedir la ejecución local de código potencialmente malicioso.

El éxito de este tipo de aplicaciones (la visión del equipo de Gosling) no fue realmente el esperado debido a diversos factores, siendo quizás el más importante la lentitud y el reducido ancho de banda de las comunicaciones en aquel entonces que limitaba el tamaño de las applets que se incrustaban en el navegador. La aparición posterior de otras alternativas (aplicaciones web dinámicas de servidor) dejó un reducido ámbito de uso para esta tecnología, quedando hoy relegada fundamentalmente a componentes específicos para la intermediación desde una aplicación web dinámica de servidor con dispositivos ubicados en la máquina cliente donde se ejecuta el navegador. Las *applets* Java no son las únicas tecnologías (aunque sí las primeras) de componentes complejos incrustados en el navegador. Otras tecnologías similares pueden ser: ActiveX de Microsoft, Flash, Java Web Start, etc.

5.5.1.3.3 En sistemas de servidor

En la parte del servidor, Java es más popular que nunca, desde la aparición de la especificación de Servlets y JSP (Java Server Pages). Hasta entonces, las aplicaciones web dinámicas de servidor que existían se basaban fundamentalmente en componentes CGI y lenguajes interpretados. Ambos tenían diversos inconvenientes (fundamentalmente lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica).



Los servlets y las JSPs supusieron un importante avance ya que:

- ✓ El API de programación es muy sencilla, flexible y extensible.
- ✓ Los servlets no son procesos independientes (como los CGI) y por tanto se ejecutan dentro del mismo proceso que la JVM mejorando notablemente el rendimiento y reduciendo la carga computacional y de memoria requeridas.
- ✓ Las JSPs son páginas que se compilan dinámicamente (o se pre-compilan previamente a su distribución) de modo que el código que se consigue una ventaja en rendimiento substancial frente a muchos lenguajes interpretados.

La especificación de Servlets y JSPs define un API de programación y los requisitos para un contenedor (servidor) dentro del cual se puedan desplegar estos componentes para formar aplicaciones web dinámicas completas. Hoy día existen multitud de contenedores (libres y comerciales) compatibles con estas especificaciones.

A partir de su expansión entre la comunidad de desarrolladores, estas tecnologías han dado paso a modelos de desarrollo mucho más elaborados con frameworks (pe Struts, Webwork) que se superponen sobre los servlets y las JSPs para conseguir un entorno de trabajo mucho más poderoso y segmentado en el que la especialización de roles sea posible (desarrolladores, diseñadores gráficos, ...) y se facilite la reutilización y robustez de código. A pesar de todo ello, las tecnologías que subyacen (Servlets y JSPs) son substancialmente las mismas.

Este modelo de trabajo se ha convertido en un estándar *de-facto* para el desarrollo de aplicaciones web dinámicas de servidor y otras tecnologías (pe. ASP) se han basado en él.

5.5.1.3.4 En aplicaciones de escritorio

Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PC de usuario de los sistemas operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecuten en cualquier PC.



En las primeras versiones de la plataforma Java existían importantes limitaciones en las APIs de desarrollo gráfico (AWT). Desde la aparición de la biblioteca Swing la situación mejoró substancialmente y posteriormente con la aparición de bibliotecas como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. sea relativamente sencillo.

5.5.1.4 Plataformas soportadas

Una versión del entorno de ejecución Java JRE (Java Runtime Environment) está disponible en la mayoría de equipos de escritorio. Sin embargo, Microsoft no lo ha incluido por defecto en sus sistemas operativos. En el caso de Apple, éste incluye una versión propia del JRE en su sistema operativo, el Mac OS. También es un producto que por defecto aparece en la mayoría de las distribuciones de Linux. Debido a incompatibilidades entre distintas versiones del JRE, muchas aplicaciones prefieren instalar su propia copia del JRE antes que confiar su suerte a la aplicación instalada por defecto. Los desarrolladores de applets de Java o bien deben insistir a los usuarios en la actualización del JRE, o bien desarrollar bajo una versión antigua de Java y verificar el correcto funcionamiento en las versiones posteriores.

5.5.1.5 Industria relacionada

Sun Microsystem, como creador del lenguaje de programación Java y de la plataforma JDK, mantiene fuertes políticas para mantener una especificación del lenguaje así como de la máquina virtual a través del JCP. Es debido a este esfuerzo que se mantiene un estándar de facto.

Son innumerables las compañías que desarrollan aplicaciones para Java y/o están volcadas con esta tecnología:

- ✓ La industria de la telefonía móvil está fuertemente influenciada por la tecnología Java.
- ✓ El entorno de desarrollo Eclipse ha tomado un lugar importante entre la comunidad de desarrolladores Java.
- ✓ La fundación Apache tiene también una presencia importante en el desarrollo de bibliotecas y componentes de servidor basados en Java.



- ✓ IBM, BEA, IONA, Oracle,... son empresas con grandes intereses y productos creados en y para Java.

5.5.1.6 Versiones del lenguaje Java

JDK 1.0 (23 de enero de 1996)

JDK 1.1 (19 de febrero de 1997). Una reestructuración intensiva del modelo de eventos AWT (Abstract Windowing Toolkit), clases internas (inner classes), JavaBeans, JDBC (Java Database Connectivity), para la integración de bases de datos, RMI (Remote Method Invocation).

J2SE 1.2 (8 de diciembre de 1998 - Nombre clave Playground. Esta y las siguientes versiones fueron recogidas bajo la denominación Java 2 y el nombre "J2SE" (Java 2 Platform, Standard Edition), reemplazó a JDK para distinguir la plataforma base de J2EE (Java 2 Platform, Enterprise Edition) y J2ME (Java 2 Platform, Micro Edition).

- ✓ La palabra reservada (keyword) strictfp.
- ✓ Reflexión en la programación.
- ✓ La API gráfica (Swing) fue integrada en las clases básicas.
- ✓ La máquina virtual (JVM) de Sun fue equipada con un compilador JIT (Just in Time) por primera vez.
- ✓ Java Plug-in.
- ✓ Java IDL, una implementación de IDL (Interfaz para Descripción de Lenguaje) para la interoperabilidad con CORBA
- ✓ Colecciones (Collections)

2SE 1.3 (8 de mayo de 2000) - Nombre clave Kestrel.

- ✓ La inclusión de la máquina virtual de HotSpot JVM (la JVM de HotSpot fue lanzada inicialmente en abril de 1999, para la JVM de J2SE 1.2)
- ✓ RMI fue cambiado para que se basara en CORBA.
- ✓ JavaSound
- ✓ Se incluyó el Java Naming and Directory Interface (JNDI) en el paquete de librerías principales (anteriormente disponible como una extensión).



- ✓ Java Platform Debugger Architecture (JPDA)

J2SE 1.4 (6 de febrero de 2002) - Nombre Clave Merlin.

Este fue el primer lanzamiento de la plataforma Java desarrollado bajo el Proceso de la Comunidad Java como JSR 59. Los cambios más notables fueron: comunicado de prensalista completa de cambios.

- ✓ Palabra reservada assert (Especificado en JSR 41.)
- ✓ Expresiones regulares modeladas al estilo de las expresiones regulares Perl.
- ✓ Encadenación de excepciones Permite a una excepción encapsular la excepción de bajo nivel original.
- ✓ Non-blocking NIO (New Input/Output) (Especificado en JSR 51.)
- ✓ Logging API (Specified in JSR 47.)
- ✓ API I/O para la lectura y escritura de imágenes en formatos como JPEG o PNG
- ✓ Parser XML integrado y procesador XSLT (JAXP) (Especificado en JSR 5 y JSR 63.)
- ✓ Seguridad integrada y extensiones criptográficas (JCE, JSSE, JAAS) Java Web Start incluido (El primer lanzamiento ocurrió en Marzo de 2001 para J2SE 1.3) (Especificado en JSR 56.)

J2SE 5.0 (30 de septiembre de 2004) - Nombre clave: Tiger. (Originalmente numerado 1.5, esta notación aún es usada internamente) Desarrollado bajo JSR 176, Tiger añadió un número significativo de nuevas características comunicado de prensa. Plantillas (genéricos) - provee conversión de tipos (type safety) en tiempo de compilación para colecciones y elimina la necesidad de la mayoría de conversión de tipos (type casting). (Especificado por JSR 14.)

- ✓ Metadatos - también llamados anotaciones, permite a estructuras del lenguaje como las clases o los métodos, ser etiquetados con datos adicionales, que puedan ser procesados posteriormente por utilidades de proceso de metadatos. (Especificado por JSR 175.)
- ✓ Autoboxing/unboxing - Conversiones automáticas entre tipos primitivos (Como los int) y clases de envoltura primitivas (Como Integer). (Especificado por JSR 201.)



- ✓ Enumeraciones - la palabra reservada enum crea una typesafe, lista ordenada de valores (como Dia.LUNES, Dia.MARTES, etc.). Anteriormente, esto solo podía ser llevado a cabo por constantes enteras o clases construidas manualmente (enum pattern). (Especificado por JSR 201.)
- ✓ Varargs (número de argumentos variable) - El último parámetro de un método puede ser declarado con el nombre del tipo seguido por tres puntos (e.g. void drawtext(String... lines)). En la llamada al método, puede usarse cualquier número de parámetros de ese tipo, que serán almacenados en un array para pasarlos al método.
- ✓ Bucle for mejorado - La sintaxis para el bucle for se ha extendido con una sintaxis especial para iterar sobre cada miembro de un array o sobre cualquier clase que implemente Iterable, como la clase estándar Collection, de la siguiente forma:
- ✓ `void displayWidgets (Iterable widgets) { for (Widget w : widgets) { w.display(); } }`
Este ejemplo itera sobre el objeto Iterable widgets, asignando, en orden, cada uno de los elementos a la variable w, y llamando al método display() de cada uno de ellos. (Especificado por JSR 201.)

Java SE 6 (11 de diciembre de 2006) - Nombre clave Mustang. Estuvo en desarrollo bajo la JSR 270. En esta versión, Sun cambió el nombre "J2SE" por Java SE y eliminó el ".0" del número de versión.[5]. Está disponible en <http://java.sun.com/javase/6/>. Los cambios más importantes introducidos en esta versión son:

- ✓ Incluye un nuevo marco de trabajo y APIs que hacen posible la combinación de Java con lenguajes dinámicos como PHP, Python, Ruby y JavaScript.
- ✓ Incluye el motor Rhino, de Mozilla, una implementación de Javascript en Java.
- ✓ Incluye un cliente completo de Servicios Web y soporta las últimas especificaciones para Servicios Web, como JAX-WS 2.0, JAXB 2.0, STAX y JAXP.
- ✓ Mejoras en la interfaz gráfica y en el rendimiento.

Java SE 7 - Nombre clave Dolphin. En el año 2006 aún se encontraba en las primeras etapas de planificación. Se espera que su desarrollo dé comienzo en la primavera de 2006, y se estima su lanzamiento para 2008.



- ✓ Soporte para XML dentro del propio lenguaje
- ✓ Un nuevo concepto de superpaquete
- ✓ Soporte para closures
- ✓ Introducción de anotaciones estándar para detectar fallos en el software.

Además de los cambios en el lenguaje, con el paso de los años se han efectuado muchos más cambios dramáticos en la librería de clases de Java (Java class library) que ha crecido de unos pocos cientos de clases en JDK 1.0 hasta más de tres mil en J2SE 5.0. APIs completamente nuevas, como Swing y Java2D, han sido introducidas y muchos de los métodos y clases originales de JDK 1.0 están desaprobadados.

5.5.2 IDE NetBeans.

5.5.2.1 Historia

NetBeans comenzó como un proyecto estudiantil en República Checa (originalmente llamado Xelfi), en 1996 bajo la tutoría de la Facultad de Matemáticas y Física en la Universidad de Charles en Praga. La meta era escribir un entorno de desarrollo integrado (IDE) para Java parecida a la de Delphi. Xelfi fue el primer entorno de desarrollo integrado escrito en Java, con su primer pre-release en 1997.

Xelfi fue un proyecto divertido para trabajar, ya que las IDEs escritas en Java eran un territorio desconocido en esa época. El proyecto atrajo suficiente interés, por lo que los estudiantes, después de graduarse, decidieron que lo podían convertir en un proyecto comercial. Prestando espacios web de amigos y familiares, formaron una compañía alrededor de esto. Casi todos ellos siguen trabajando en NetBeans.

Tiempo después, ellos fueron contactados por Roman Stanek, un empresario que ya había estado relacionado con varias iniciativas en la República Checa. Él estaba buscando una buena idea en que invertir, y encontró en Xelfi una buena oportunidad. Ellos se reunieron, y el negocio surgió.

El plan original era desarrollar unos componentes JavaBeans para redes. Jarda Tulach, quien diseñó la arquitectura básica de la IDE, surgió con la idea de llamarlo NetBeans, con el fin de describir lo que ellos harían. Cuando las especificaciones de los



Enterprise JavaBeans salieron, ellos decidieron trabajar con este estándar, ya que no tenía sentido competir con él, sin embargo el nombre de NetBeans se quedó.

“En la primavera de 1999, Netbeans DeveloperX2 fue lanzado, soportando Swing. Las mejoras de rendimiento que llegaron con el JDK 1.3, lanzado en otoño de 1999, hicieron a NetBeans una alternativa realmente viable para el desarrollo de herramientas. En el verano de 1999, el equipo trabajó duro para rediseñar a DeveloperX2 en un NetBeans más modular, lo que lo convirtió en la base de NetBeans hoy en día.

Algo más paso en el verano de 1999. Sun Microsystems quería una mejor herramienta de desarrollo de Java, y comenzó a estar interesado en NetBeans. En otoño de 1999, con la nueva generación de NetBeans en Beta, el acuerdo fue realizado.

Sun adquirió otra compañía de herramientas al mismo tiempo, Forté, y decidió renombrar NetBeans a Forté for Java. El nombre de NetBeans desapareció de vista por un tiempo.

Seis meses después, se tomó la decisión de hacer a NetBeans open source. Mientras que Sun había contribuido considerablemente con líneas de código en varios proyectos de código abierto a través de los años, NetBeans se convirtió en el primer proyecto de código abierto patrocinado por ellos. En Junio del 2000 NetBeans.org fue lanzado”.²⁰

5.5.2.2 NetBeans IDE

El IDE NetBeans es un IDE - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos

²⁰ <http://www.netbeans.org/about/history.html>



de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

La versión actual es NetBeans IDE 5.5.1, la cual fue lanzada en Mayo de 2007. NetBeans IDE 5.5 extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++.

Modularidad. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Desde Julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL).

5.5.3 Gestor de Base de Datos MySQL

5.5.3.1 Historia del proyecto

SQL (*Lenguaje de Consulta Estructurado*) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que **MySQL** cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael Widenius en la década de los 90 trató de usar *mSQL* para conectar las tablas usando rutinas de bajo nivel ISAM, sin embargo, mSQL no era rápido y flexible para sus necesidades. Esto lo conllevó a crear una API SQL denominada **MySQL** para bases de datos muy similar a la de *mSQL* pero más portable.



La procedencia del nombre de MySQL no es clara. Desde hace más de 10 años, las herramientas han mantenido el prefijo My. También, se cree que tiene relación con el nombre de la hija del cofundador Monty Widenius quien se llama *MySql*.

Por otro lado, “el nombre del delfín de MySQL es Sakila y fue seleccionado por los fundadores de **MySQL** AB en el concurso “Name the Dolphin”. Este nombre fue enviado por Ambrose Twebaze, un desarrollador de Open source Africano, derivado del idioma SiSwate, el idioma local de Swazilandia y corresponde al nombre de una ciudad en Arusha, Tanzania, cerca de Uganda la ciudad origen de Ambrose”.²¹

5.5.3.2 Lenguajes de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

5.5.3.3 Aplicaciones

“MySQL es muy utilizado en aplicaciones web como MediaWiki, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones”.²²

²¹ <http://es.wikipedia.org/wiki/MySQL>

²² <http://www.mysql.com/>



5.5.3.4 Especificaciones

Características:

Entre sus principales características se destacan en:

➤ Lenguajes de programación: Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi, Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

- ✓ “Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- ✓ Transacciones y claves foráneas.
- ✓ Replicación.
- ✓ Búsqueda e indexación de campos de texto.
- ✓ Soporte a multiplataforma.
- ✓ Uso de multihilos mediante hilos del kernel.
- ✓ Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice.
- ✓ Tablas hash en memoria temporales.
- ✓ Completo soporte para operadores y funciones en cláusulas select y where.
- ✓ Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación.
- ✓ Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- ✓ Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- ✓ Se permiten hasta 64 índices por tabla. Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes.



- ✓ Procedimientos almacenados.
- ✓ Triggers.
- ✓ Cursores.
- ✓ Vistas actualizables.
- ✓ Soporte a VARCHAR.
- ✓ INFORMATION_SCHEMA.
- ✓ Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle.
- ✓ Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial).
- ✓ Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB.
- ✓ Sub-SELECTs (o SELECTs anidados).
- ✓ Soporte completo para Unicode²³

5.5.3.5 Plataformas

MySQL funciona sobre múltiples plataformas, incluyendo:

- ✓ "AIX
- ✓ BSD
- ✓ FreeBSD
- ✓ HP-UX
- ✓ GNU/Linux
- ✓ Mac OS X
- ✓ NetBSD
- ✓ Novell Netware
- ✓ OpenBSD
- ✓ OS/2 Warp
- ✓ QNX
- ✓ SGI IRIX
- ✓ Solaris
- ✓ SunOS

²³ <http://es.wikipedia.org/w/index.php> "Mysql"



- ✓ SCO OpenServer
- ✓ SCO UnixWare
- ✓ Tru64
- ✓ Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.
- ✓ OpenVMS²⁴.

5.5.4 Geotools

5.5.4.1 Introducción

Geotools es una librería orientada al desarrollo de aplicaciones SIG sobre plataformas Java. Provee métodos que permiten manipular la información geográfica obtenida y además dibujar las geometrías vectoriales.

Actualmente están disponibles dos versiones llamadas Geotools 1 (GT1) y Geotools 2 (GT2) las cuales presentan características distintas.

La primera (GT1) tiene como objetivo manipular archivos shape, al igual que la segunda (GT2) por razones de compatibilidad, además ésta última está orientada a la conectividad de datos bajo OpenGis. Ambas librerías están actualmente en continuo desarrollo, los fuentes, binarios, documentación y demos se encuentran disponibles en www.geotools.org.

5.5.4.2 Geotools 1 (GT1)

Está compuesta por tres paquetes:

- geotools.jar
- collections.jar
- support.jar

Además es posible encontrar algunos ejemplos sencillos sobre su uso y algunos mapas para comenzar a programar:

²⁴[http:// www.pi-net.dyndns.org/anonymous/kits/](http://www.pi-net.dyndns.org/anonymous/kits/).



<http://multilingual.geotools.org/resources/>

Geotools 1 está orientado al uso de archivos shapefiles (*.shp) que se usa en Arcview. Por lo que también incluye soporte para los archivos *.dbf que contienen información referente a las geometrías.

Las librerías que forman parte de este paquete son aproximadamente un 80% más livianas que las usadas en Geotools 2. En las pruebas realizadas, y disponibles en www.inf.uct.cl/~flara/prototipos.htm se ve claramente que la demo desarrollada con GT1 pesa 775 KB y la demo realizada con GT2 pesa 7,92 MB (incluidas las librerías gráficas JavaJAI). Lo que hace pensar que GT1 es más portable sobre internet en aplicaciones como Applet por el menor tiempo de descarga que esto implica.

5.5.4.3 Geotools 2 (GT2)

Es un gran conjunto de librerías, de las cuales solo están disponibles los fuentes. El proceso de compilación es algo simple, pero puede transformarse en algo extremadamente complejo si no se tiene una idea clara de los pasos a seguir. Una vez compilados se obtienen 30 archivos JAR de los cuales 11 son necesarios para generar cualquier tipo de aplicación:

- ✓ core-SNAPSHOT.jar
- ✓ cts-coordtrans-SNAPSHOT.jar
- ✓ data-SNAPSHOT.jar
- ✓ defaultcore-SNAPSHOT.jar
- ✓ gui-SNAPSHOT.jar
- ✓ lite-rendering-SNAPSHOT.jar
- ✓ opengis-SNAPSHOT.jar
- ✓ resources-SNAPSHOT.jar
- ✓ shapefile-SNAPSHOT.jar
- ✓ sldstyling-SNAPSHOT.jar
- ✓ utils-SNAPSHOT.jar



Además, se requiere de tres paquetes adicionales para que los paquetes anteriores funcionen, estos son parte del repository de maven y se obtienen durante el proceso de compilación:

- ✓ JTS-1.2.jar
- ✓ JTS-1.3.jar
- ✓ Vecmath-1.2.1.jar

GT2 presenta una característica más que la hace diferente de la GT1, y es que utiliza para dibujar las geometrías, librerías de manejo de imagen de java llamadas JavaJAI que se pueden obtener en el sitio de SunMicrosystem (<http://java.sun.com/products/java-media/jai/current.html>).

5.5.4.4 Geotools2 para la Gestión de Mapas

GeoTools2 son un conjunto de librerías “open Source” para desarrollar soluciones GIS. Tiene una arquitectura modular que permite añadir nuevas funcionalidades de manera muy sencilla y presume de soportar Open GIS y otros estándares de desarrollo para sistemas geográficos. Está construido utilizando el último JDK de Sun (1.4.1) y se mantiene continuamente actualizado para aprovechar las mejoras que van saliendo en la plataforma Java. Además está dividido en diferentes módulos, cada uno de ellos implementa un requerimiento específico. Normalmente, solo algunos de estos módulos son necesarios para crear una aplicación gis.

Geotools2 en interacción con el sistema

En la siguiente figura (Figura 17) podemos apreciar la interacción de geotools2 en el resto del sistema:

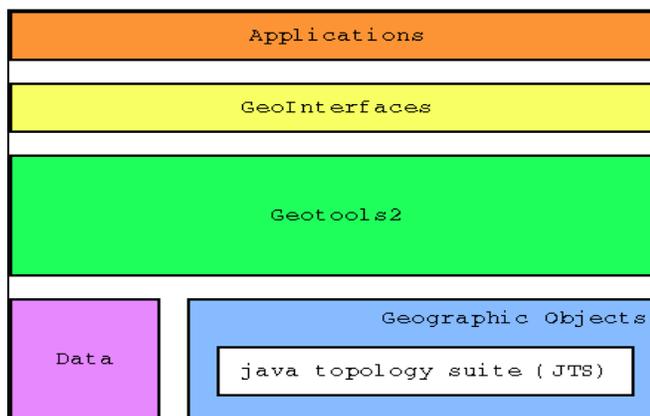


FIGURA 17: Geotools 2 En Contexto



Nivel de aplicaciones: Son aplicaciones geográficas que usan primitivas de geotools2. Normalmente se usan para producir visores de mapas o “web map servers”

Nivel de geolnterfaces: Contiene interfaces para todas las primitivas geográficas, como “Features” (objeto geográfico) o “Data Source” (Almacén de datos espaciales).

Nivel de data: Son los objetos geográficos que maneja geotools2, pueden venir de base de datos (Postgis), ESRI shapefiles o cualquier otro formato geográfico

Nivel de JTS: Es una librería de Java que proporciona todas las primitivas para tratar con objetos geográficos como puntos, polígonos o líneas.

5.5.4.5 Diseño de Geotools2

Los Distintos módulos de geotools2 se pueden resumir en la figura 14 (Módulos de Geotools2):

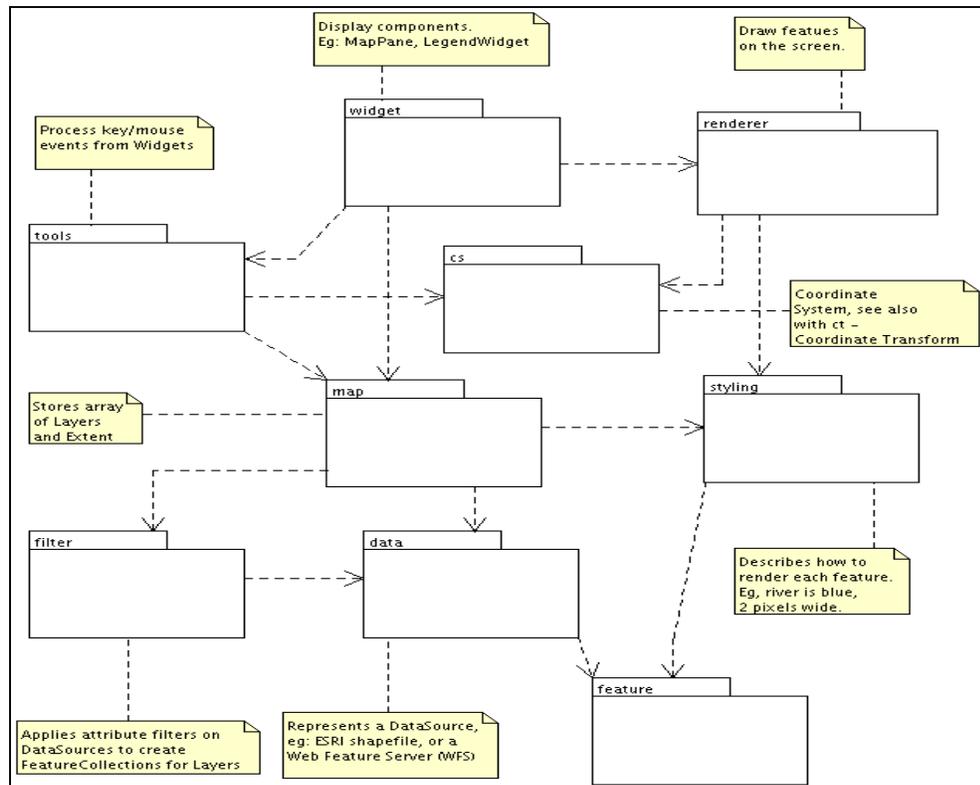


FIGURA 18. Módulos de Geotools 2



Widget: Es un componente que muestra algo en la pantalla. Por ejemplo un MapPane, para ver un mapa, un LegendWidget o un PanLeftButton.

Renderer: Es el responsable de dibujar las “Features” en la pantalla. Se llama desde Map.

Styling: Especifica que determinadas “Features” serán dibujadas acorde a una serie de atributos.

Map: Almacena una lista de capas y una caja asociada al mapa, esta caja tiene la misma dimensión que el conjunto de “Features” contenidas en el mapa.

Tools: Procesan los eventos de ratón y teclado en el mapa y que causan cambios en él. Como ejemplo tenemos el ZoomTools, que realiza un zoom determinado sobre una parte del mapa.

Cs y ct: Son el sistema de coordenadas y la transformación de las mismas.

Data: Es la interface con numerosos “almacenes de datos”. Como los Shapefiles de ESRI, o una base de datos Postgis.

Filter: Los filtros se aplican a los “almacenes de datos” para crear una lista de “Features” determinada en una capa. Un ejemplo sería:

Dime los contenedores donde el tipo = Vidrio.

Feature: Contiene las bases geográficas de las “Features”: Líneas, polígonos, puntos, etc



5.5.4.6 Comparación con otras librerías

A continuación se presenta un cuadro con las funcionalidades de un SIG frente a los distintos tipos de librerías existentes:

FUNCIONALIDADES	Geo Tools		Map Server		Map Window		Gis Viewer		Map Guide	
	B	10	B	10	B	10	B	10	B	10
Navegación y visualización dinámica e interactiva	B	10	B	10	B	10	B	10	B	10
Selección de elementos	B	10	B	10	B	10	B	10	B	10
Control de visualización según detalle	B	10	B	10	B	10	B	10	B	10
Consultas gráficas y lógicas	B	10	X	6	B	10	B	10	B	10
Operaciones geométricas básicas(corredor, distancias)	N	0	N	0	P	5	N	0	B	10
Creación de elementos gráficos temporales	B	10	N	0	N	0	N	0	P	8
Agregación de capas adicionales	N	0	B	10	B	8	N	0	P	8
Actualización de datos geográficos y atributos.	N	0	B	5	B	8	N	0	P	4
Generación de reportes sobre consultas	N	0	B	5	P	8	N	0	B	5
Personalización del Idioma	N	0	P	4	X	5	P	0	B	5
Interacción con Páginas Web	B	5	B	5	N	0	B	5	B	5
Acceso directo a Base de Datos	B	10	B	10	B	10	P	8	B	10
TOTAL		65		75		84		53		95
B= Funcionalidad Básica (10-5), P= programable(8-4), X=Limitada(6-2), N=No Implementada(0)										

Cuadro 7. Funcionalidades del SIG frente a las librerías



En el cuadro anterior (Cuadro 7) existe un rango que permite calificar las librerías cuando se las evalúa frente a la funcionalidad de los SIGs, así por ejemplo la Navegación y visualización dinámica e interactiva, tiene una Funcionalidad Básica (B) de 10 para todas las librerías.

En lo que se refiere a nuestra aplicación hemos escogido las **librerías Geotools2**, debido a que están orientadas al desarrollo de aplicaciones SIG sobre plataformas Java. Como podemos ver en el cuadro de funcionalidades (Cuadro 7), estas librerías tienen muchos beneficios pero así mismo tienen bastantes limitaciones; es por ello que lo que no se ha podido hacer directamente con las librerías está programado con código que reemplaza a los procesos que se deberían hacer con las librerías.



6. EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN



EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN:

El objetivo de investigación es la aspiración, el propósito, (el para qué), se desarrolla la investigación que presupone al objeto transformado: es la situación propia del problema superado. Para llevar a cabo el cumplimiento detallado de los objetivos específicos planteados en el proyecto que son el objeto de investigación, fueron necesarios una serie de parámetros. A continuación se detalla cada uno de los objetivos específicos con su respectiva evaluación:

1.- Sistematizar, Integrar y Organizar la información aislada sobre los diferentes desastres naturales que ocurren en la Provincia de Loja.

Para llevar a cabo el cumplimiento de este objetivo primeramente se realizó una reunión con los directivos del CINFA (entidad que apoya el desarrollo de la aplicación) y los estudiantes encargados de recolectar la información de campo (coordenadas exactas, damnificados, muertos heridos, pérdidas, etc.), misma que fue entregada para luego ser analizada y organizada, de manera que se encuentre lista para ingresarla en el sistema desarrollado.

2.- Diseñar y Construir una Base de Datos (MySQL) que maneje la información relevante (tipo de desastre, ubicación, fecha, coordenadas, imágenes, etc.)

Con la información que se obtuvo en la etapa de recolección se procedió a diseñar una base de datos consistente donde se pueda almacenar de manera confiable la información clave y más importante que puede tener un desastre natural generado en la Provincia de Loja, de manera que pueda interactuar correctamente con el sistema. La construcción de la base de datos se la realizó con información real y precisa, mediante un gestor de base de datos de MySQL que es de distribución gratuita y muy eficiente.

3.- Proveer la información precisa de desastres más frecuentes que se presentan en lugares específicos de la Provincia de Loja.

La información que se administra en la base de datos se encuentra debidamente clasificada, de tal manera que el usuario puede tener referencia de la misma, realizando consultas sobre desastres naturales generados en lugares específicos (cantones, parroquias), además con la información almacenada se podrá realizar filtrados múltiples



con respecto a un rango de fechas, por el tipo de desastre natural generado (derrumbes, sequías, aludes, inundaciones, etc.). De esta manera se puede garantizar que la información se encuentra correctamente almacenada y distribuida en el sistema.

4.- Elaborar interfaces para el manejo, migración y estructuración de la información almacenada con miras al acoplamiento de la misma con el SIG y a la interacción con el usuario.

Para mayor comodidad del usuario del sistema se construyeron ventanas de visualización gráfica completamente amigables y adecuadas que garantizan una mejor interacción del usuario con la aplicación, de manera que se pueda tener un manejo más sencillo y eficiente de la aplicación. Para ello se tomaron muy en cuenta los requerimientos del usuario recogidos uno a uno durante la etapa de análisis.

5.- Diseñar y Construir el SIG utilizando tecnología JAVA, en base a la información obtenida sobre los desastres naturales que ocurren en la Provincia de Loja.

La etapa de diseño y construcción es aquella en la que se realiza la codificación del software de la aplicación. El código se realizó utilizando el lenguaje JAVA mediante el IDE de desarrollo NetBeans, teniendo en cuenta que este lenguaje es de distribución libre, por lo que no hay costos adicionales de licencias. Para la manipulación de archivos shape (capas de la provincia de Loja) se utilizó el conjunto de librerías Geotools2, mismas que proveen una serie de métodos necesarios para el manejo de estos archivos y para trabajar con geometrías vectoriales. Los reportes que se presentan de cada desastre generado fueron elaborados utilizando las librerías JasperReport y el plugin de Ireport que también se incorporan en IDE de NetBeans.

6.- Implementar el SIG en el Centro Integrado de Geomática Ambiental (CINFA) de la UNL.

Una vez concluido el sistema bajo todos los parámetros explicados anteriormente, se procede a implantarlo en la entidad que nos apoyó con sus conocimientos y nos guió durante todo el desarrollo del proyecto; mediante un instalador completo que incluye todas las funcionalidades que el usuario había sugerido, de manera que se da cumplimiento a todos los requerimientos planteados al inicio del proyecto.



7. DESARROLLO DE LA PROPUESTA ALTERNATIVA



7.1 Definición del Problema

La Universidad Nacional de Loja es una institución que desde hace mucho tiempo viene ofreciendo sus servicios a la colectividad lojana, a través de de sus diferentes Áreas y carreras universitarias. En el Área Agropecuaria específicamente se encuentra el Centro Integrado de Geomática Ambiental (CINFA), que en un principio fue creado debido a la poca utilización de herramientas computacionales para facilitar la toma de decisiones en la investigación y docencia por parte de los docentes y demás profesionales afines; hoy en día es uno de los centros que más aportan en el manejo de datos geográficos en sistemas de computo electrónico en el sur del país

“El objetivo general del CINFA es capturar, procesar, analizar y generar, en formato análogo o digital información geográfica relacionada con los recursos biofísicos (agua, suelo, aire, flora y fauna) y socioeconómicos, utilizando tecnología computarizada”.²⁵

En este mismo centro existen personas encargadas de recolectar la información de campo como lo son los desastres naturales, pero en la actualidad todos estos datos solo se almacenan en forma manual (cálculos manuales en hojas Excel, información en hojas de Word) ya que no existe un software especializado en donde se puedan visualizar y manipular todos los desastres naturales ocurridos en la provincia de Loja, solo existen programas como el ArcGis que son generales y que solo pueden ser manejados por especialistas debido a que requieren demasiado entrenamiento; haciéndose necesario e imprescindible elaborar un software especializado que cumpla con todas las necesidades y requerimientos referentes a ubicación y clasificación de desastres naturales..

Por consiguiente y debido a que no existe en la ciudad de Loja un software de estas características, esta investigación pretende **diseñar y desarrollar un SIG para la ubicación y clasificación de desastres naturales en la Provincia de Loja**, que permita a los directivos del CINFA y autoridades de la Defensa Civil elaborar un plan de contingencia y tomar decisiones acertadas para evitar futuros desastres, y de esta manera estaríamos contribuyendo al desarrollo y superación de la sociedad y la población del sur

²⁵ http://www.cinfa.edu.ec/Acerca_de_CINFA



del país; todo ello bajo la asesoría técnica de especialistas y profesionales que laboran en el CINFA.

Para ello hemos recogido uno a uno los requerimientos del usuario que es quien va a estar en contacto directo y va a interactuar con el sistema. Estos requerimientos son los siguientes:

REQUERIMIENTOS DEL USUARIO	
1.	Integrar la información aislada en hojas de Excel y Word.
2.	Interfaz amigable y fácil de utilizar.
3.	Respuesta rápida e interacción directa con el sistema
4.	Que permita visualizar los mapas de la provincia y de cada cantón, así como sus parroquias, ríos, vías.
5.	Que se muestre información general de la provincia y sus cantones
6.	Que permita realizar consultas generales y específicas de los desastres naturales registrados
7.	Que se pueda ingresar nuevos mapas
8.	Que se pueda ingresar nuevos desastres
9.	Que permita presentar reportes de cada desastre
10.	Que permita imprimir los reportes relacionados con la información del desastre natural generado.
11.	Que permita visualizar e imprimir fotografías correspondientes a cada desastre
12.	Que se pueda hacer acercamientos, alejamientos, giros de los mapas.

CUADRO 8: Requerimientos del Usuario

Teniendo en cuenta los requerimientos anteriores, decidimos plantear como solución el “Sistema de Información Geográfica para la Ubicación y Clasificación de Desastres Naturales en la Provincia de Loja” que permita manipular e ingresar toda la información referente a desastres naturales de una manera fácil, rápida e interactiva, por lo que se propuso utilizar herramientas open source y que sean de fácil uso. Estas herramientas son las siguientes:

- ✓ Lenguaje de programación JAVA utilizando el IDE NetBeans.
- ✓ Base de Datos MySQL.
- ✓ Librerías Geotools2 y JasperReport
- ✓ Pluggin de IReport.

La siguiente figura muestra la arquitectura de la aplicación ya desarrollada, la cual define la funcionalidad externa del sistema.

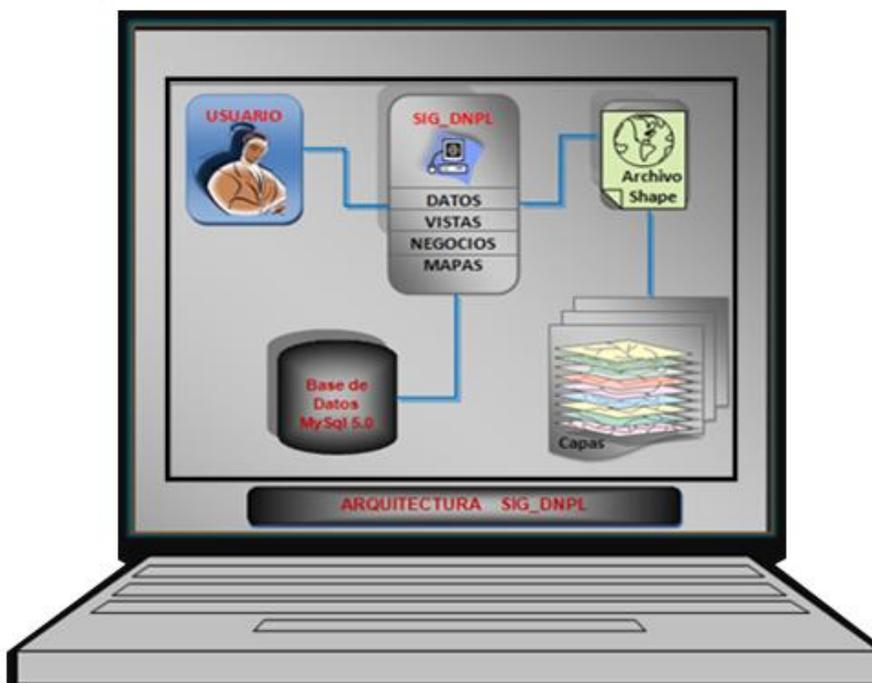


FIGURA 19: Arquitectura SIG_DNPL

El estudio de los desastres naturales que se generan en la provincia de Loja comienza con la clasificación de los diferentes tipos de desastres que se pueden dar, entre los que tenemos: sismos, tsunamis, erupciones volcánicas, derrumbes, aluviones, aludes, huaycos, inundaciones, sequías, heladas, tormentas, granizadas, huracanes, tornados, plagas, epidemias. Además se debe tener en cuenta la ubicación exacta de cada cantón y parroquia de la Provincia.

Con el conocimiento de toda esta información ya podemos apuntar directamente al proyecto cuyo objetivo es ubicar y clasificar desastres naturales en el mapa de la Provincia de Loja, teniendo en cuenta las coordenadas como referencia para ubicar los puntos.

Al referirnos a la ubicación y clasificación quiere decir que se harán todos los procedimientos necesarios para integrar la información de desastres naturales que reposa



en el CINFA, de manera que sirva como soporte a los directivos para la toma de decisiones.

El desarrollo del proyecto incluye el cálculo de “coordenadas”²⁶ exactas (latitud, longitud) de cada punto en el mapa, así como el número de damnificados, muertos y heridos, pérdidas ocasionadas, importancia, ubicación (cantón, parroquia), efectos, etc.

En lo que se refiere al tipo de coordenadas existen 2 tipos: “coordenadas planas o UTM”²⁷ (dadas en metros) y coordenadas geográficas (dadas en grados, minutos y segundos). Los mapas solo soportan coordenadas planas, por lo que cuando las coordenadas son de tipo geográficas, es necesario convertirlas a planas, y para ello se siguió el siguiente procedimiento (Ver Figura 19) que para un mejor entendimiento se va a explicar mediante un ejemplo.

7.1.1 Pasos a seguir para la conversión de Coordenadas

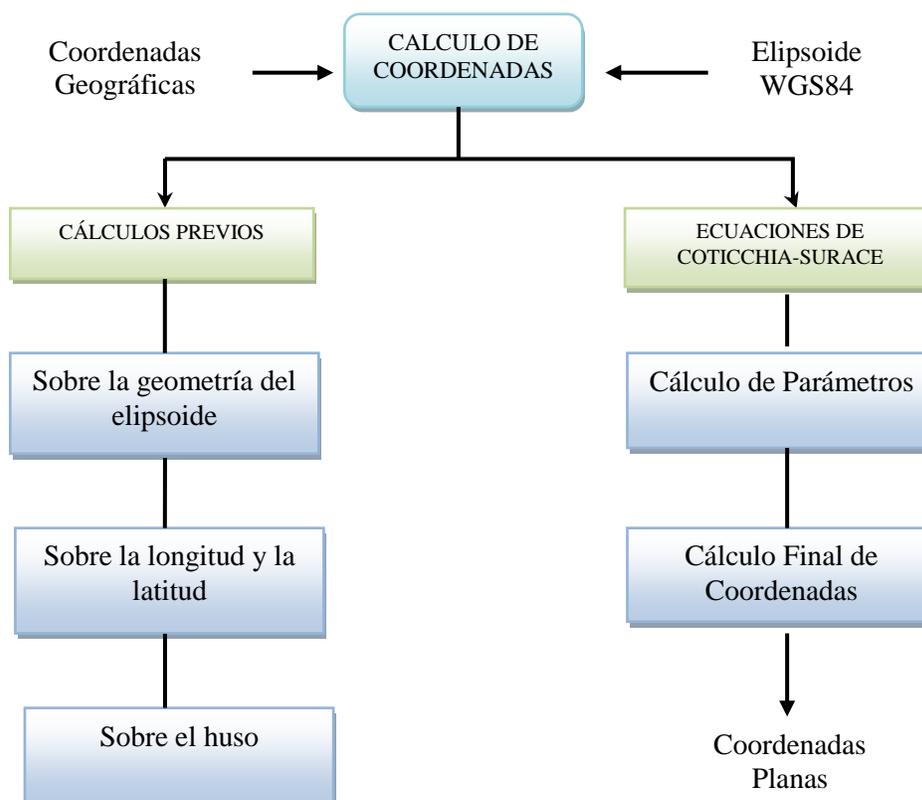


Figura 20. Esquema de conversión de Coordenadas

²⁶ Se dice de las líneas que sirven para determinar la posición de un punto, y de los ejes a que se refieren aquellas líneas.

²⁷ Universal Transversal Mercator.



Partimos en primer lugar de las coordenadas geográficas del vértice con el que haremos el ejemplo, es decir necesitamos las coordenadas dadas en grados, minutos y segundos. Los datos de este vértice están en principio en geográficas sobre el “elipsoide WGS84”²⁸. Las coordenadas de tipo geográficas son:

“Longitud”²⁹: $\lambda = 79^\circ 38' 21''$

“Latitud”³⁰: $\varphi = 4^\circ 9' 55''$

También vamos a necesitar los datos básicos de la geometría del elipsoide, es decir el semieje mayor (a) y al semieje menor (b). A partir de estos datos, aprenderemos a deducir otros parámetros de la geometría del elipsoide que nos harán falta en el proceso de conversión de coordenadas. Así, los datos referentes a los “semiejes”³¹ del elipsoide WGS84 son:

Semieje Mayor (a) = 6378137

Semieje Menor (b) = 6356752,314

Cabe recalcar que estos valores son fijos y únicos para el elipsoide WGS84. Con estos datos ya podemos empezar a operar. En negro se indicarán las ecuaciones originales y en azul los datos correspondientes al desarrollo del ejemplo.

Los cálculos que se realizan a continuación son operaciones previas al cálculo final de coordenadas que se realizará más adelante mediante las ecuaciones de Cotichia-Surace utilizadas para el ejemplo. Estos cálculos son los siguientes:

7.1.1.1 Sobre la geometría del “elipsoide”³²

Primero calculamos la excentricidad, la segunda excentricidad, el radio polar de curvatura y el aplanamiento para lo que se emplean las siguientes fórmulas:

²⁸ World Geodetic System 1984 (Sistema Geodésico Mundial 1984).

²⁹ Magnitud física que expresa la distancia entre dos puntos. Su unidad en el Sistema Internacional es el metro.

³⁰ Distancia que hay desde un punto de la superficie terrestre al Ecuador, contada en grados de meridiano.

³¹ Representan el radio de la Elipsoide (Semieje mayor y Semieje menor)

³² Superficie de aproximación a la forma de la tierra empleada por los diferentes sistemas cartográficos, en el caso de Ecuador nuestro elipsoide de referencia es el WGS84.



$$\text{Excentricidad } e = \frac{\sqrt{a^2 - b^2}}{a} \longrightarrow e = \frac{\sqrt{6378137^2 - 6356752,314^2}}{6378137} = 0,08181919$$

$$\text{Segunda Excentricidad } e' = \frac{\sqrt{a^2 - b^2}}{b} \longrightarrow e' = \frac{\sqrt{6378137^2 - 6356752,314^2}}{6356752,314} = 0,08209443$$

Aprovechamos para calcular también el cuadrado de la segunda excentricidad, pues nos hará falta en muchos pasos posteriores, es decir e'^2 :

$$e'^2 = 0,08209443 = 0,00673949$$

Seguimos con el radio polar de curvatura y el aplanamiento:

$$\text{Radio Polar de Curvatura: } c = \frac{a^2}{b} \longrightarrow c = \frac{6378137^2}{6356752,314} = 6399593,626$$

$$\text{Aplanamiento: } \alpha = \frac{a-b}{a} \longrightarrow \alpha = \frac{6378137 - 6356752,314}{6378137} = 0,00335281$$

En realidad, el aplanamiento y la excentricidad (la primera excentricidad), no son necesarios para la aplicación de las "ecuaciones de Cotichia-Surace"³³, pero están incluidas porque frecuentemente los parámetros del elipsoide se dan como el semieje mayor (a) y el aplanamiento (α), o bien como el semieje mayor (a) y la excentricidad (e). En estas circunstancias, conociendo las correspondientes fórmulas podríamos también calcular el parámetro del semieje menor (b).

7.1.1.2 Sobre la Longitud y la Latitud:

A continuación veremos las fórmulas para calcular la longitud y la latitud, lo primero que hacemos es convertir los grados sexagesimales (grados, minutos y segundos) a grados sexagesimales expresados en notación decimal (lo que se suele denominar normalmente "grados decimales"). Para ello operamos de la siguiente forma:

³³ Ecuaciones utilizadas para el paso directo de coordenadas geográficas a planas



Grados decimales = grados + minutos/60 + segundos/60/60

$\left\{ \begin{array}{l} \text{La longitud (79}^\circ \text{ 38' 21''), pasaría a ser en sexadecimales expresados} \\ \text{en notación decimal} = 79 + 38/60 + 21/60/60 = -79.63916666666667 \\ \\ \text{La latitud (4}^\circ \text{ 9' 55''), pasaría a ser en sexadecimales expresados} \\ \text{en notación decimal} = 4 + 9/60 + 55/60/60 = -4.1652777777777778 \end{array} \right.$

Una vez que tenemos la longitud y la latitud en grados decimales, procedemos a su paso a radianes, pues la mayor parte de los pasos posteriores se realizarán con entrada de datos en radianes. Operamos para ello de la forma:

$$\text{Radianes} = \frac{\text{Grados decimales} * \pi}{180}$$

$\left\{ \begin{array}{l} \text{La longitud (-79.63916666666667), pasaría a ser en} \\ \text{radianes} = -79.63916666666667 * \pi / 180 = -1.3899656718778508 \\ \\ \text{La latitud (-4.1652777777777778), pasaría a ser en} \\ \text{radianes} = -4.1652777777777778 * \pi / 180 = -0.07269781148237492 \end{array} \right.$

El siguiente paso es calcular el signo de la longitud. Para ello el proceso lógico es muy sencillo:

$\left\{ \begin{array}{l} \text{Si la longitud está referida al este del} \\ \text{meridiano de Greenwich, entonces la} \\ \text{longitud es positiva (+)} \\ \\ \text{Si la longitud está referida al oeste} \\ \text{del meridiano de Greenwich,} \\ \text{entonces la longitud es negativa (-)} \end{array} \right. \rightarrow \left\{ \begin{array}{l} \text{Como la longitud está referida al} \\ \text{oeste del meridiano de} \\ \text{Greenwich, entonces la longitud} \\ \text{es negativa, quedando en} \\ \text{radianes} = - \\ 1.3899656718778508 \text{ y en} \\ \text{grados decimales} = - \\ 79.63916666666667 \end{array} \right.$

7.1.1.3 Sobre el “Huso”³⁴:

Una vez tenemos preparados los datos de longitud y latitud, podemos calcular el huso o zona UTM (*UTM Zone*) donde caen las coordenadas a convertir, con las siguientes operaciones:

$$\text{Huso} = \text{entero de} \left[\frac{\text{Grados decimales}}{6} + 31 \right] \rightarrow \text{Huso} = \left[\frac{-79.63916666666667}{6} + 31 \right] = 17.72680$$

³⁴ Sub-región en que divide el planeta un sistema de proyección cartográfico para mantener sus propiedades métricas.

Truncamos la parte entera, luego Huso = 17

Con el huso ya conocido, el siguiente paso es obtener el meridiano central de dicho huso. El meridiano central es la línea de tangencia del cilindro transverso.

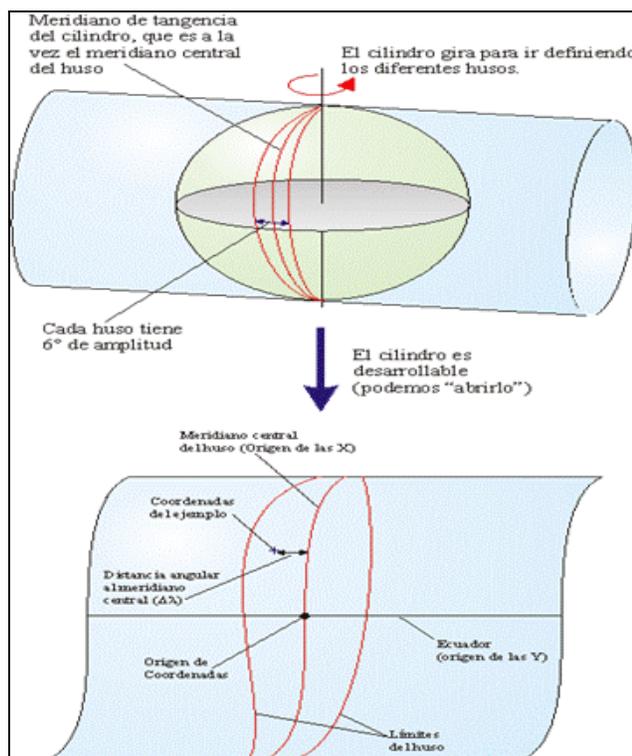


Figura 21. Representación del meridiano central del huso o zona

Pero antes de seguir con los cálculos e introducir más conceptos, vamos a repasar algunos de los elementos principales de la proyección UTM. Así, conviene recordar que en la proyección UTM el cilindro transverso que se usa como superficie desarrollable, se va girando virtualmente para definir los diferentes husos (60) que rodean la tierra.

Se empiezan a contar los husos por el antimeridiano de Greenwich, así por ejemplo la Provincia de Loja cae en el huso 17.

El meridiano central del huso es muy importante porque es el origen de las coordenadas X. Como el meridiano central dejaría la parte del huso situada a su izquierda con coordenadas X negativas, para evitar eso, se suma a todas las coordenadas X la



cantidad de 500.000. Esto hace que no existan valores negativos para las coordenadas X, puesto que se ha realizado un retranqueo del eje X de 500 km

Y algo semejante se hace para los valores de Y, cuyo origen es el ecuador. Si el ecuador es el origen de las Y, toda la parte situada al sur del mismo tendría coordenadas negativas. Para evitar eso, se suma el valor 10.000.000 a los valores de Y, pero sólo en el caso de que se trate de coordenadas pertenecientes al hemisferio sur; si las coordenadas pertenecen al hemisferio norte, no se tocan los valores Y.

Volviendo con el meridiano central del huso, éste también tiene la particularidad de que es automecoico. En teoría, para cualquier latitud que caiga dentro del rango de operación de la proyección UTM (intervalo entre los 84° N y los 80° S), el punto de menor deformación de la proyección UTM es el que para esa latitud se sitúa sobre el meridiano central de su correspondiente huso. En la práctica esto no es del todo cierto, pues la proyección UTM aplica un factor de escala (0,9996) que hace que las zonas de menor deformación pasen a ser las situadas a $\pm 2^{\circ} 15'$ (aproximadamente a 180 km del meridiano central, aunque esta medida varía con la latitud); son las llamadas líneas isométricas, derivadas de la aplicación de este factor de escala (denominado K0) que es una de las principales diferencias entre la Proyección UTM y la Proyección Gauss-Krüger, en la que se basa la UTM en su totalidad.

Expuestos estos conceptos, para saber mínimamente lo que estamos calculando, vamos a retomar los cálculos donde los habíamos dejado. Habíamos dicho que el siguiente paso es obtener el meridiano central del huso en el que caen las coordenadas geodésicas sobre las que operamos. La operación es muy sencilla:

$$\lambda_0 = \text{huso} * 6 - 183 \quad \longrightarrow \quad \lambda_0 = 17 * 6 - 183 = -81$$

Ahora calculamos la distancia angular que existe entre la longitud del punto con el que operamos y el meridiano central del huso. Es muy importante señalar que ambos datos tienen que ser introducidos en radianes. La longitud ya la habíamos traducido a radianes antes, pero no así el valor del meridiano central que acabamos de calcular. Para convertirlo a radianes multiplicamos por π y dividimos por 180:



$$\Delta\lambda = \lambda - \lambda_0 \longrightarrow \Delta\lambda = -1.3899656718778508 - (-81 * \pi/180) = 0.023751023$$

7.1.1.4 Ecuaciones de Coticchia-Surace para el Problema Directo (Paso de Geográficas a Planas o UTM).

a) Cálculo de Parámetros:

A continuación debemos calcular una serie de parámetros que van encadenados unos a otros y que son el núcleo de las ecuaciones de Coticchia-Surace. Son muchas operaciones pero como se puede ver el proceso es muy rutinario y fácilmente programable:

$$A = \cos \varphi * \sin \Delta\lambda \longrightarrow A = (\cos -0.07269781148) * (\sin 0.023751023) = 0.02368606105$$

$$\xi = \frac{1}{2} * \ln \left[\frac{1+A}{1-A} \right] \longrightarrow \xi = \frac{1}{2} * \ln \left[\frac{1+0.023686061057823227}{1-0.023686061057823227} \right] = 0.02369049207571745$$

$$\eta = \tan^{-1} \left(\frac{\tan \varphi}{\cos \Delta\lambda} \right) - \varphi \longrightarrow \eta = \tan^{-1} \left(\frac{\tan -0.072697}{\cos 0.02375} \right) - (-0.0726978) = -2.043742434E - 5$$

$$v = \frac{c}{(1+e'^2 * \cos^2 \varphi)^{1/2}} * 0.9996 \longrightarrow v = \frac{6399593.626}{(1+0.00673949 * \cos^2 -0.0726978114)^{1/2}} * 0.9996 = 6375698.332$$

$$\zeta = \frac{e'^2}{2} * \xi^2 * \cos^2 \varphi \longrightarrow \zeta = \frac{0.00673949}{2} * (0.0236904)^2 * \cos^2(-0.072697) = 1.8812580E - 6$$

$$A_1 = \text{sen}(2 * \varphi) \longrightarrow A_1 = \text{sen}(2 * -0.07269781148) = -0.14488388965263413$$

$$A_2 = A_1 * \cos^2 \varphi \longrightarrow A_2 = -0.144883889 * \cos^2 -0.0726978114 = -0.14411953034597$$

$$J_2 = \varphi + \frac{A_1}{2} \longrightarrow J_2 = -0.07269781148 + \frac{-0.14488388965263}{2} = -0.14513975630869197$$

$$J_4 = \frac{3 * J_2 + A_2}{4} \longrightarrow J_4 = \frac{3 * (-0.1451397563086) + (-0.144119530345)}{4} = -0.14488469981801202$$



$$J_6 = \frac{5 \cdot J_4 + A_2 \cdot \cos^2 \varphi}{3} \longrightarrow J_6 = \frac{5 \cdot (-0.144884699) + (-0.144119530) \cdot \cos^2 - 0.0726978114}{3} = -0.2892609008$$

$$\alpha = \frac{3}{4} \cdot e'^2 \longrightarrow \alpha = \frac{3}{4} \cdot 0,00673949 = 0.0050546226149520476$$

$$\beta = \frac{5}{3} \cdot \alpha^2 \longrightarrow \beta = \frac{5}{3} \cdot 0.0050546226149520476^2 = 4.256498349278807E - 5$$

$$\gamma = \frac{35}{27} \cdot \alpha^2 \longrightarrow \gamma = \frac{35}{27} \cdot 0.0050546226149520476^2 = 1.6736753115567135E - 7$$

$$B_\phi = 0,9996 \cdot c \cdot (\varphi - \alpha J_2 + \beta J_4 - \gamma J_6)$$

$$B_\phi = 0,9996 \cdot 6399593,626 \\ * (-0.07269 - (0.00505 \cdot -0.14513) + (4.25649834E - 5 \cdot -0.14488469) \\ - (1.673675311E - 7 \cdot -0.2892609)) = -460396.4625569821$$

b) Cálculo Final de Coordenadas

Una vez que disponemos de todos los parámetros anteriores calculados, procedemos a la solución de las coordenadas UTM finales, de la siguiente forma:

$$X = \xi \cdot v \cdot \left(1 + \frac{\zeta}{3}\right) + 500000$$

$$X = 0.023690 \cdot 6375698.332 \cdot \left(1 + \frac{1.8812580E-6}{3}\right) + 500000 = 651043.525540$$

Solución de la X UTM

Como la Provincia de Loja se encuentra situada a la izquierda del Meridiano de Greenwich, lo que arroja coordenadas X negativas, existe la necesidad de sumar el valor de 500000 al resultado, lo que permitirá evitar como resultado estos valores negativos.

Para el caso de la solución de Y es muy importante recordar que si la latitud de las coordenadas geodésicas con las que operamos pertenece al hemisferio sur deberemos sumar el valor 10.000.000 al resultado obtenido.



$$Y = (\eta * v * (1 + \zeta) + B_{\phi}) + 10000000$$

Y =

$$-2.043742434E - 5 * 6375698.332 * (1 + 1.8812580E - 6) + (-460396.4625569821) + 10000000 = 9539470.147652015$$

Solución de la Y UTM

Al igual que en la solución de X, la Provincia de Loja se encuentra en el Hemisferio Sur, por lo que existe la necesidad de sumar el valor de 10000000 al resultado obtenido.

7.2 DOCUMENTACIÓN DE REQUERIMIENTOS

A continuación, se explican los aspectos que enmarcan el análisis y documentación de requerimientos del “SIG para la Ubicación y Clasificación de Desastres Naturales en la Provincia de Loja”.

7.2.1 Requerimientos Funcionales

El sistema está orientado a proveer de una interfaz de usuario por medio de la cual permita visualizar y obtener información relevante sobre los desastres naturales ocurridos en la Provincia de Loja y tomar decisiones a partir de los mismos.

Además, debe proveer de las herramientas que permitan realizar acciones comunes sobre los datos como lo son:

REQUERIMIENTOS FUNCIONALES	
Ref. Req.	Descripción
Rf001	Disponer de una barra de herramientas que sea de fácil interpretación y manipulación de todas las opciones que dispone el sistema.
Rf002	Permitir instalar la base de datos creada para el efecto directamente desde el sistema ubicada en una dirección específica del ordenador.
Rf003	Clasificar en la BD todos los Desastre Naturales con sus respectivos tipos, y su correspondiente visualización en



	componentes gráficos.
Rf004	Realizar una descripción breve al momento de seleccionar un determinado tipo de Desastre Natural, para conocer sobre que se trata.
Rf005	Permitir actualizar o insertar un determinado tipo de Desastre Natural.
Rf006	Asignar un color representativo para cada tipo de desastre Natural, por parte del usuario.
Rf007	Guardar en la BD nuevos archivos Shape “capas” que se desean visualizar permanentemente. Con ayuda de un navegador de directorios para obtener su dirección.
Rf008	Realizar actualizaciones de los archivos Shape disponibles en BD.
Rf009	Fijar un color propio que represente a cada archivo Shape al momento de representarlo en el mapa.
Rf010	Controlar que el usuario solo pueda eliminar archivos Shape de tipo anexo y no de tipo base.
Rf011	Visualizar el mapa base (capas) de la provincia de Loja (cantones, parroquias, cabeceras parroquiales, ríos, vías, alturas, bosques, etc), incluido la capa de desastres naturales, la misma que se representa como puntos.
Rf012	Manipular la lista de capas base, de acuerdo a lo que el usuario quiera visualizar, ya sea agregando o eliminando capas, de manera temporal.
Rf013	Refrescar la lista de capas para volver a un estado inicial, en caso de existir algún problema en la manipulación de los mapas “capas”.
Rf014	Definir un elemento leyenda que muestre, si la capa seleccionada es de tipo polígono, punto o línea y que lo dibuje con su color representativo.
Rf015	Manejar los mapas ó capas base con sus respectivos desastres, a través del menú de navegación que contiene



	acciones de zoom, enfoques, desplazamientos, etc.
Rf016	Listar los cantones de la Provincia Loja para ubicarlos gráficamente, resaltando su posición dentro del mapa.
Rf017	Realizar consultas directamente a la capa desastres generados del mapa, es decir que al momento de hacer clic en un punto pueda mostrar la información de dicho desastre generado en la provincia de Loja.
Rf018	Visualizar en una interfaz apropiada la información específica de cada desastre natural, como lo es: nombre, tipo, parroquias y población afectada, número de damnificados, heridos y muertos, localización, importancia, costo de pérdida, fecha de ocurrencia y efectos, etc. Con ayuda de un menú de acceso, o desde el mapa según el usuario lo requiera.
Rf019	Crear un reporte completo sobre los datos que se registran de un determinado desastre, que el usuario visualiza.
Rf020	Visualizar un Informe General con todos los datos adicionales, que han sido detallados y redactados por el usuario.
Rf021	Crear un reporte sobre el Informe General del desastre, si el usuario lo solicita.
Rf022	Visualizar fotografías reales y específicas de cada desastre mediante un guía de imágenes.
Rf023	Crear un reporte de la imagen del desastre que el usuario requiera.
Rf024	Actualizar ó Eliminar la información existente en la base de datos sobre los desastres naturales ocurridos, y modificar la representación grafica en el mapa.
Rf025	Ingresar un nuevo desastre a la BD con toda la información que el usuario disponga, a que cantón pertenece y la respectiva representación gráfica como punto en el mapa teniendo como referencia las coordenadas geográficas correspondientes.
Rf026	Redactar un Informe General del Desastre con datos adicionales que el usuario maneje y que no han sido



	considerados anteriormente, según su criterio.
Rf027	Agregar imágenes al nuevo desastre desde cualquier dirección en donde se encuentren, con ayuda de un navegador de directorios.
Rf028	Proporcionar al usuario la posibilidad convertir coordenadas geográficas a Planas "UTM", con ayuda de componentes gráficos, para que puedan ser aceptadas por el sistema.
Rf029	Realizar un control minucioso de cada uno de los datos que el usuario ingresa, al momento de Guardar o Actualizar, para que conozca si existe alguna inconsistencia en la compatibilidad de los mismos y corrija su error.
Rf030	Realizar consultas múltiples de la información disponible en la BD, tomando en consideración aspectos importantes como, rangos de fecha, cantones, tipos de desastre natural, etc. Y su visualización dentro de una tabla la misma en la que se define campos de mayor importancia.
Rf031	Generar reportes de los datos filtrados que se encuentran disponible en la Tabla de resultados, previo a una selección del usuario en la columna Reportes.
Rf032	Visualizar los datos filtrados que se muestran en la tabla ya sea en el mapa de la Provincia o del Cantón, según el usuario lo requiera, para visualizar la ubicación física de cada desastre.
Rf033	Realizar consultas desde la tabla de datos filtrados, en busca de un determinado desastre con toda la información que le corresponde, realizando un clic sobre la fila respectiva.
Rf034	Almacenar el último filtrado realizado, para hacer uso de este según sea requerido por el usuario, en caso de perder los datos consultados.
Rf035	Disponer de una opción que muestre una lista de todos los desastres filtrados, y al seleccionar un elemento de la lista este se resalte en el mapa para conocer su ubicación, cuyo color debe representar al tipo de desastre que le pertenece.



Rf036	Mostrar datos políticos e información general de la Provincia de Loja y de cada uno de sus cantones.
Rf037	Facilitar información que permita describir de una manera general lo que es un SIG, para que el usuario conozca más sobre este tema y su importancia.
Rf038	Proveer una ventana de ayuda sobre el manejo del programa para que el usuario pueda utilizar de manera eficiente el sistema.

Cuadro 9. Requerimientos Funcionales

7.2.2 Requerimientos no Funcionales

Los requerimientos no funcionales se describen a continuación, estos son aspectos que no forman parte de la aplicación pero que se encuentran presentes en la misma:

REQUERIMIENTOS NO FUNCIONALES		
Atributo	Ref. Req.	Descripción
Documentación	Rfn001	La documentación será orientada principalmente a los programadores, es decir, entregará detalles técnicos comprensibles solo para otros desarrolladores. Detallará los métodos y clases usadas en el desarrollo, de forma de entregar un mayor nivel de detalle para usuarios interesados en ampliar o mejorar las opciones del SIG.
Consideraciones de Software	Rfn001	El sistema trabaja bajo las herramientas del lenguaje JAVA de la plataforma NetBeans.
	Rfn002	El sistema utiliza una base de datos generada en



		Mysql.
	Rfn003	El sistema trabaja con las librerías Geotools, para la gestión de mapas.
	Rfn004	El Sistema trabaja con el Plugin de IReport y las librerías JasperReport para generar reportes
Interfaz	Rfn004	El sistema estará orientado a una fácil manipulación de sus componentes físicos como: Cuadros de diálogo, mapas cartográficos y sus controles, etc. que interactúan directamente con el usuario.
Validación	Rfn005	El sistema se encuentra validado en el Centro Integrado de Geomática Ambiental (CINFA) de la UNL.
Consideraciones de Hardware	Rfn006	Las aplicaciones Java tienen la particularidad de correr en cualquier computador con Java Virtual Machine (JVM) instalado. Por lo que no se requiere un hardware específico, pero si es necesario un computador con buenas prestaciones como Sistema Operativo Microsoft Windows XP o Vista y Memoria RAM de 1 GB o superior.

Cuadro 10. Requerimientos No Funcionales



7.3 GLOSARIO DE TÉRMINOS

ArcView:

El producto de software más liviano de la familia ArcGIS.

ArcGIS:

Software para el manejo de información geográfica producido por ESRI.

Coordenadas:

Se dice de las líneas que sirven para determinar la posición de un punto, y de los ejes o planos a que se refieren aquellas líneas.

Datum:

Un sistema geodésico local queda definido por la elección de un elipsoide de referencia y por un punto origen (Datum), donde se establece su ubicación en relación con la forma física de la tierra (geoide). Concretamente el punto Datum es aquel en el que se hace coincidir la vertical del lugar con la normal al Elipsoide (desviación de la vertical nula), y generalmente se establece la conexión de tangencia entre el elipsoide y el geoide.

Elipsoide:



“Superficie de aproximación a la forma de la tierra empleada por los diferentes sistemas cartográficos. Se trata de la mejor forma posible de describir el geoide en términos matemáticos. Dada la complejidad de la forma de la tierra, es imposible tratarla tal como es en realidad con matemáticas, por lo que es necesario reducir su forma principal a una forma geométrica susceptible de ser descrita con números. El elipsoide es precisamente esa figura de aproximación que permite aplicar los diferentes sistemas de proyección cartográficos”.³⁵

Huso o zona:

Sub-región en que divide el planeta un determinado sistema de proyección cartográfico para mantener sus propiedades métricas.

Isoterma:

Es una curva que une los puntos, en un plano cartográfico, que presentan las mismas temperaturas en la unidad de tiempo considerada. Así, para una misma área, se pueden diseñar un gran número de planos con isotermas, por ejemplo: Isotermas de la temperatura media de largo periodo del mes de enero, de febrero, etc., o las isotermas de las temperaturas medias anuales.

Latitud:

Distancia que hay desde un punto de la superficie terrestre al Ecuador, contada en grados de meridiano.

Longitud:

Magnitud física que expresa la distancia entre dos puntos. Su unidad en el Sistema Internacional es el metro.

Mapa

Es un conjunto de puntos, líneas y áreas que están definidas tanto por su localización en el espacio con referencia a un sistema de coordenadas como por sus atributos no espaciales.

³⁵ www.gabrielortiz.com/diccionariocartografico

**Shape:**

Son archivos que almacenan por un lado datos geométricos y por otros atributos de la información espacial en un conjunto de datos. La geometría de un objeto es almacenada como un "shape" el cual comprende un conjunto de coordenadas vectoriales. Los "ShapeFile" soportan los tipos básicos de elementos, es decir el punto, la línea y el polígono. Los atributos descriptivos son almacenados en un archivo dBase. Cada atributo tiene asociado una relación de uno a uno con un registro de un "shape" asociado

SIG (GIS)

Sistema de Información Geográfica. Son sistemas computarizados compuestos de software y procedimientos que permiten capturar, almacenar, manipular, gestionar, analizar, modelar y presentar datos referenciados espacialmente. Estos sistemas no solo almacenan información acerca de la localización de los elementos en el espacio (georeferenciación), sino también acerca de las relaciones entre unos elementos y otros (topología).

Sistema WGS84:

El acrónimo WGS84 deviene de Word Geodetic System 1984 (Sistema Geodésico Mundial 1984). Se trata de un sistema de referencia creado por la Agencia de Mapeo del Departamento de Defensa de los Estados Unidos de América (DMA) para sustentar la cartografía producida en dicha institución y las operaciones del departamento de Defensa. Este sistema está estrechamente ligado al desarrollo del Sistema de Poesionamiento Global (GPS)

UML

Es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede utilizarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web. (Método de Lenguaje Unificado).

UTM:



Cilíndrica horizontal; conforme el meridiano Central y el Ecuador son líneas rectas, el resto de los meridianos y paralelos son líneas complejas, la escala es verdadera a lo largo de 2 líneas paralelas al meridiano central (secante), con un factor de escala igual a 1. Al meridiano central se le asigna un factor de escala igual a 0,9996; husos de 6 grados de ancho, a las coordenadas del hemisferio sur se le suma 10 millones de metros, evitándose con esto las coordenadas negativas. El meridiano central tiene una coordenada X igual a 500 mil metros



7.4 Modelado

7.4.1 Modelo de Dominio



7.4.2 Modelo de Casos de Uso

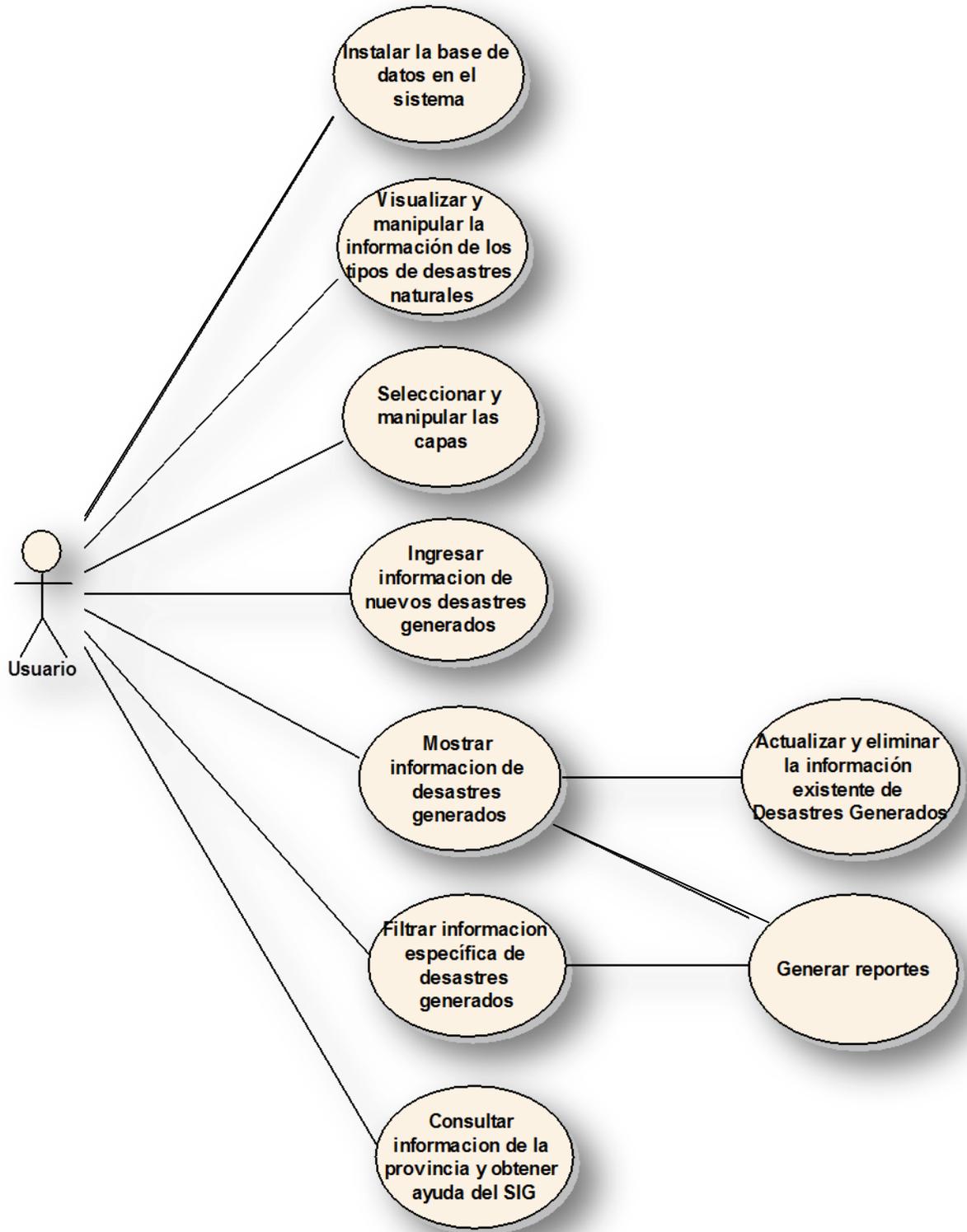


FIGURA 23: Casos de Uso



7.4.2.1 Caso de Uso # 1: Instalar la base de datos en el sistema



FIGURA 24: UC01

CASO DE USO :	Instalar la base de datos en el sistema
Identificador:	UC01
Ref. Req.:	Rf002
Propósito:	Instalar la base de datos desde una ubicación específica del ordenador
Descripción:	Antes de arrancar el sistema por primera vez, el usuario tiene la opción de instalar la base de datos que el sistema necesita para cargar correctamente todos sus componentes
Tipo:	Primario Real.
Actor:	Usuario.
Precondición:	El sistema debe estar instalado en el computador El usuario debe seleccionar el acceso directo "SIG_DNPL" para ingresar al sistema El usuario debe estar en la Ventana de Inicio del sistema El usuario debe visualizar la etiqueta "Iniciar"
Curso normal de eventos:	<ol style="list-style-type: none"> 1. El usuario da un clic sobre la etiqueta "INICIAR" que se encuentra en la Ventana de Inicio 2. El sistema muestra un mensaje que indica que la base de datos aún no se encuentra instalada. 3. El usuario acepta el mensaje 4. El sistema muestra la ventana "Instalar Base de Datos" con las opciones para ejecutar la base de datos desde la ubicación



	<p>donde se encuentra el archivo por defecto.</p> <ol style="list-style-type: none"> 5. El usuario selecciona el botón “EJECUTAR” para instalar la base de datos 6. El sistema lee el archivo gis_desastre.sql desde la ubicación recomendada 7. El sistema muestra un mensaje en la parte inferior de la ventana que indica que la base de datos ha sido instalada 8. El usuario selecciona el botón “SALIR” 9. El sistema cierra de la ventana de instalación de la base de datos y muestra otra vez la Ventana de Inicio 10. El usuario selecciona la etiqueta “INICIAR” por segunda vez 11. El sistema carga los componentes necesarios para su correcto funcionamiento 12. El caso de Uso finaliza.
<p>Curso alternativo de eventos:</p>	<p>A. Instalar BD desde directorio</p> <ol style="list-style-type: none"> A5. El usuario selecciona la opción “Directorio” A6. El sistema muestra la ventana con las opciones para ejecutar la base de datos desde cualquier ubicación. A7. El usuario navega en el directorio donde se encuentre el archivo sql. A8. El sistema carga el archivo desde la ubicación que el usuario indique. A9. El caso de uso continúa en el paso 5 del Curso Normal de Eventos.
<p>Post Condiciones:</p>	<ul style="list-style-type: none"> ➤ Relacionar la información almacenada en la base de datos con el sistema de una manera eficiente. ➤ Cargar todos los componentes básicos del sistema en la ventana principal.

CUADRO 11. Caso de Uso # 1



✓ Diagrama de Robustez

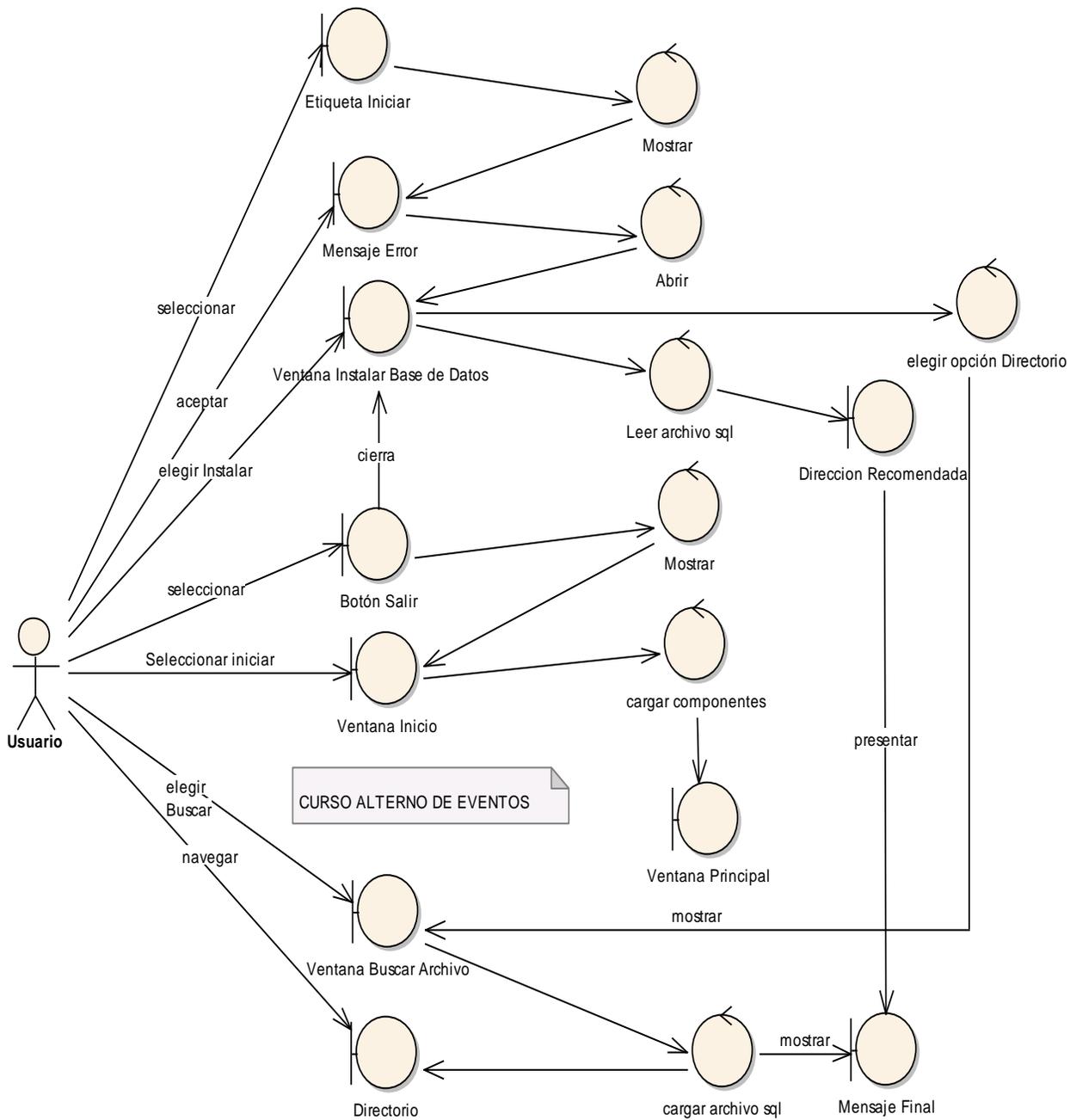


FIGURA 25: Diagrama de robustez # 1



✓ Diagrama de Secuencia

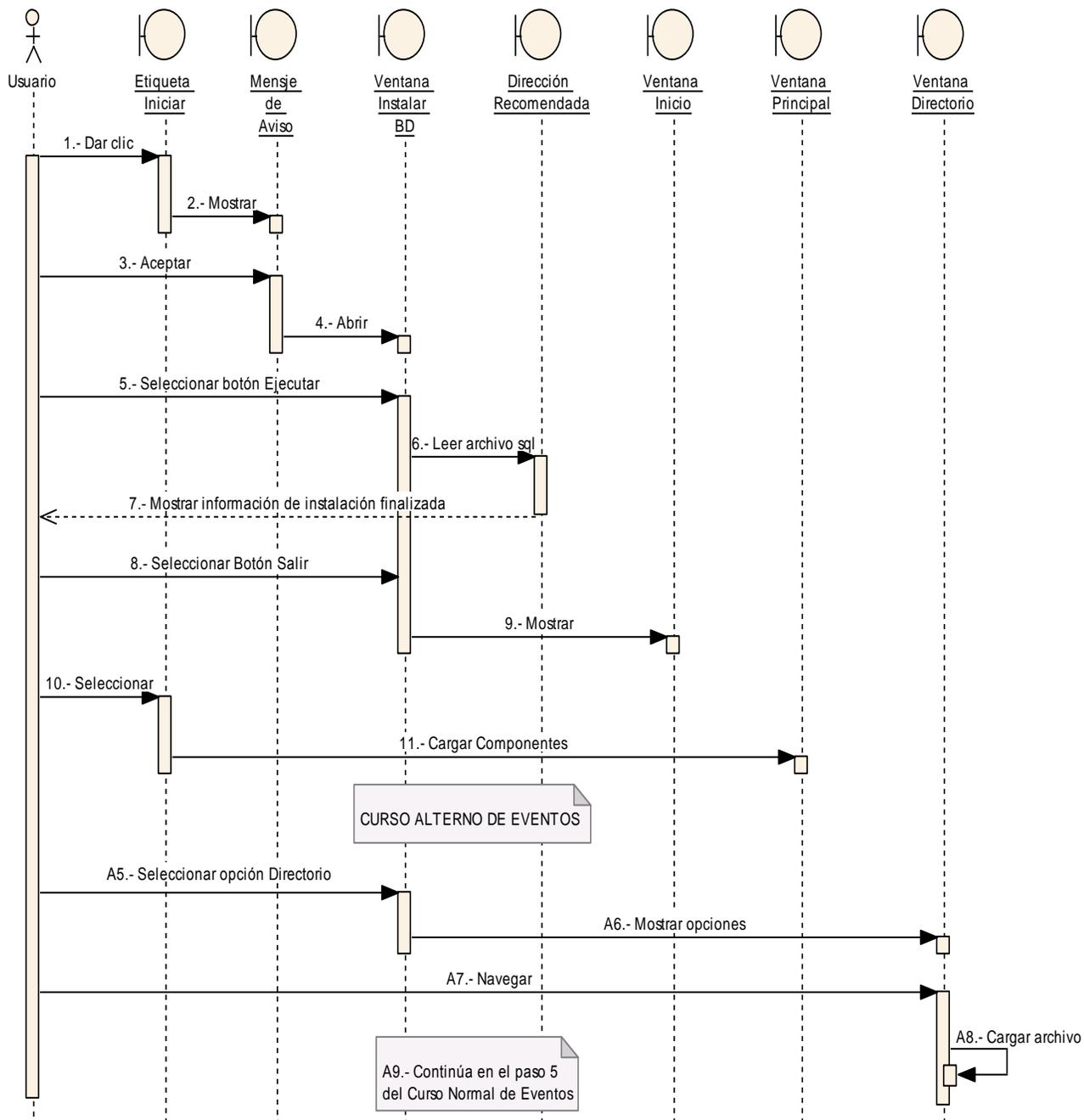


FIGURA 26: Diagrama de secuencia # 1



7.4.2.2 Caso de Uso # 2: Visualizar y manipular la información de los tipos de desastres naturales

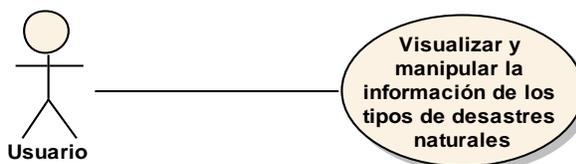


FIGURA 27: UC02

CASO DE USO :	Visualizar y manipular la información referente a los tipos de desastres naturales
Identificador:	UC02
Ref. Req.:	Rf001, Rf002, Rf003, Rf004, Rf005
Propósito:	Observar la clasificación de Desastres Naturales existentes y manipular (insertar, actualizar, eliminar) los tipos de desastre natural
Descripción:	El usuario puede observar los desastres naturales clasificados de acuerdo a su tipo, y además puede ingresar nueva información y actualizar la existente
Tipo:	Primario Real.
Actor:	Usuario.
Precondición:	El usuario debe estar en la Ventana Principal El usuario debe visualizar el menú "Edición" El sistema debe estar conectado con la base de datos creada para el efecto en MySQL.
Curso normal de eventos:	<ol style="list-style-type: none"> 1. El usuario ingresa al menú Edición, luego a Archivo y luego a Desastre Natural 2. El sistema muestra una ventana con la información de Desastres Naturales. 3. El usuario selecciona un tipo de desastre natural y activa el botón "Insertar" 4. El sistema muestra la ventana Insertar desastre natural con



	<p>los campos vacíos y editables.</p> <ol style="list-style-type: none"> 5. El usuario ingresa el nombre del desastre natural 6. El usuario selecciona el botón Definir Color Leyenda 7. El sistema muestra la paleta de colores 8. El usuario selecciona el color de la leyenda y acepta el nuevo color 9. El sistema muestra el nuevo color en la leyenda del desastre natural 10. El usuario ingresa la descripción del tipo de desastre natural 11. El usuario presiona el botón GUARDAR 12. El sistema guarda los cambios en la Base de Datos. 13. El caso de Uso finaliza.
<p>Curso alterno de eventos:</p>	<p>A. Actualizar Desastre Natural</p> <p>A3. El usuario selecciona un desastre natural y activa el botón “Actualizar”</p> <p>A4. El sistema genera la Ventana Modificar Desastre Natural</p> <p>A5. El usuario realiza los cambios</p> <p>A6. El caso de uso continúa en el paso 11 del Curso Normal de Eventos.</p> <p>B. Eliminar Desastre Natural</p> <p>B4. El usuario selecciona un desastre natural y activa el botón “Eliminar”</p> <p>B5. El sistema muestra un mensaje de confirmación para eliminar un desastre natural</p> <p>B6. El usuario confirma la petición</p> <p>B7. El sistema borra el desastre natural por completo de la base de datos</p> <p>B8. El caso de uso finaliza</p>
<p>Post Condiciones:</p>	<ul style="list-style-type: none"> ➤ Visualizar la clasificación de desastres naturales ➤ Actualizar un desastre natural ➤ Eliminar un desastre natural de la base de datos ➤ Ingresar un nuevo tipo desastre natural

CUADRO 12. Caso de Uso # 1



✓ Diagrama de Robustez

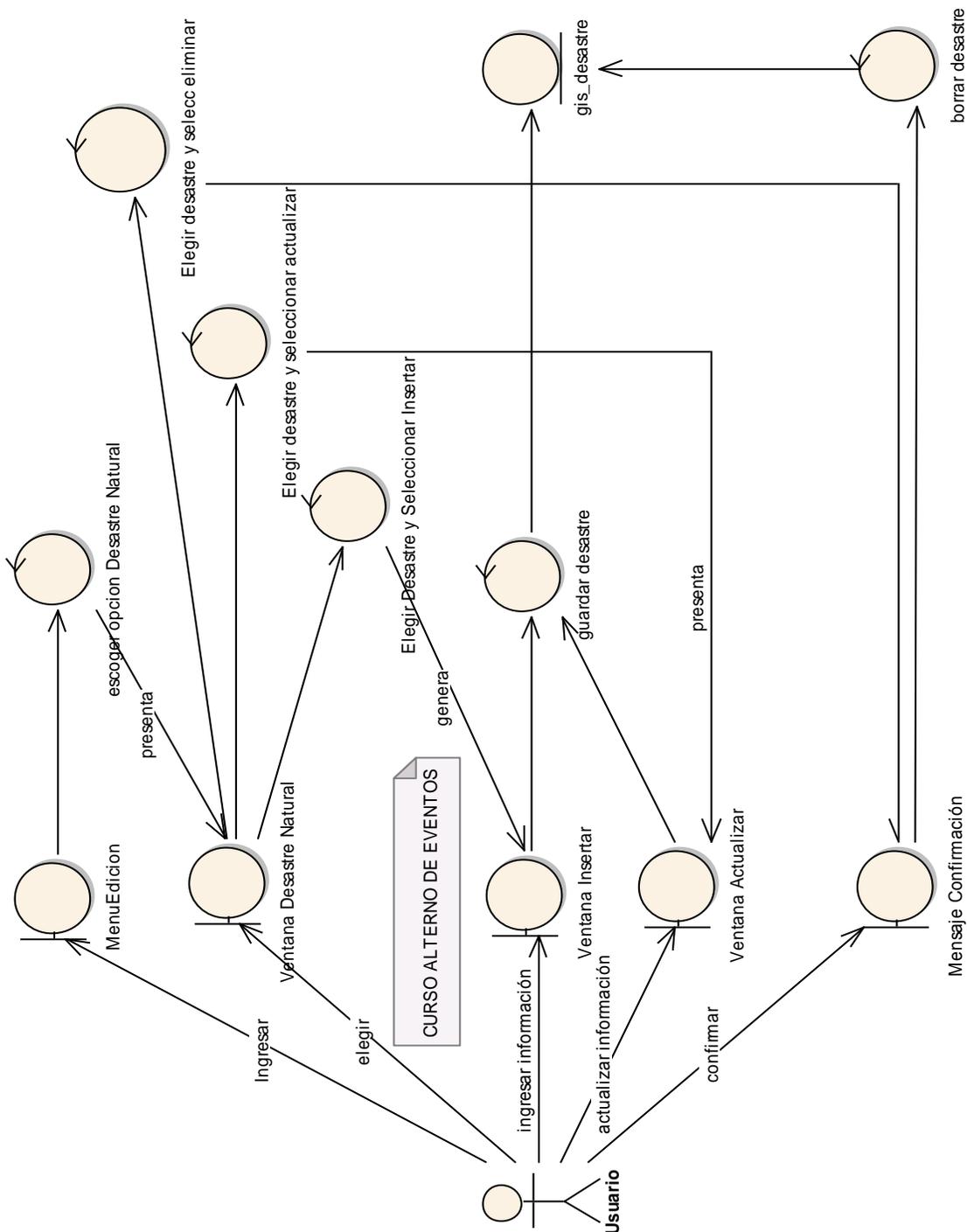


FIGURA 28: Diagrama de robustez # 2



✓ Diagrama de Secuencia

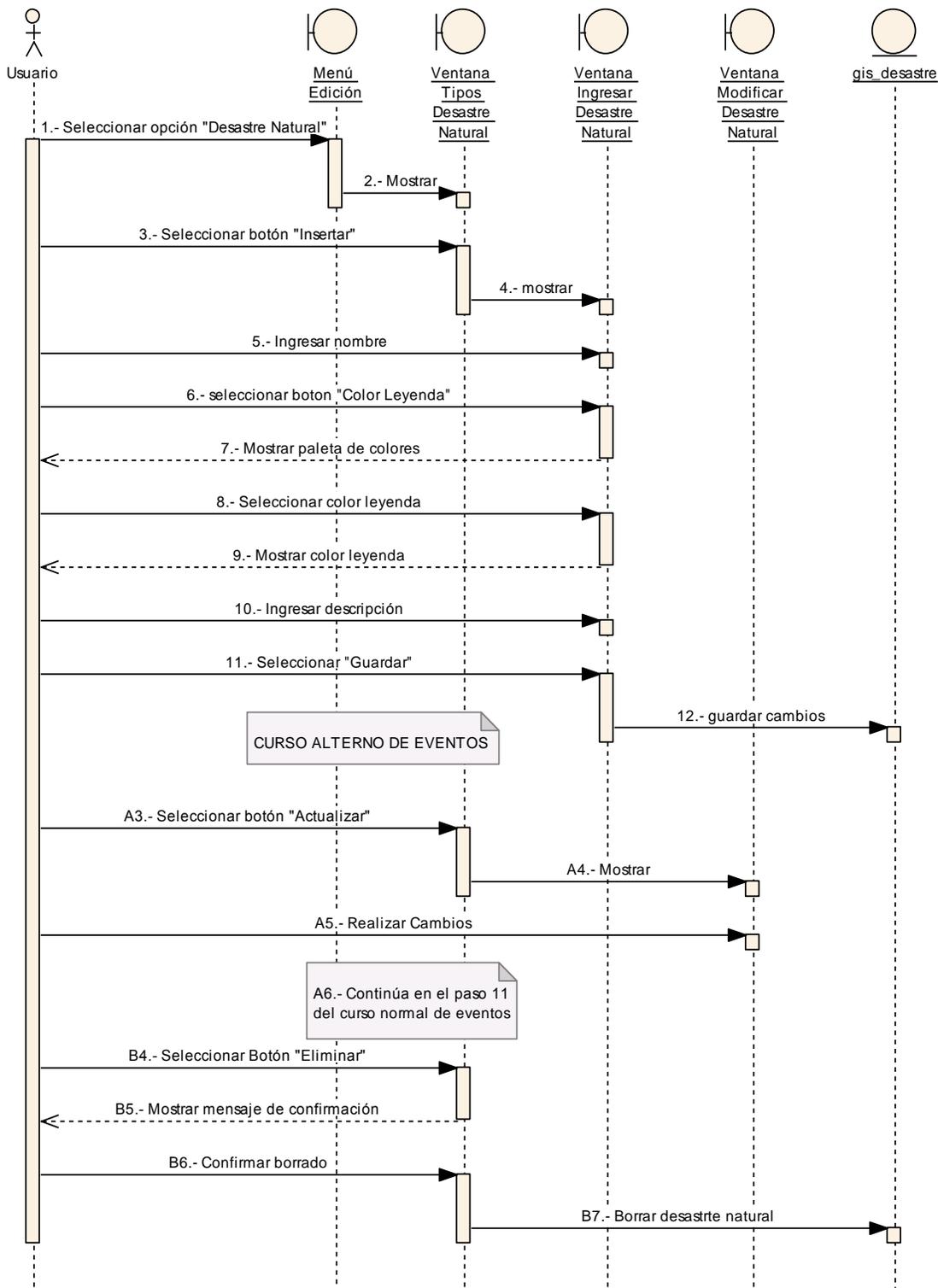


FIGURA 29: Diagrama de Secuencia # 2

7.4.2.3 Caso de Uso # 3: Seleccionar y manipular las capas

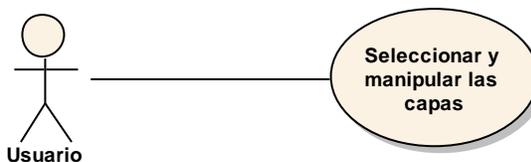


FIGURA 30: UC03

CASO DE USO :	Seleccionar y manipular las capas
Identificador:	UC03
Ref. Req.:	Rf006, Rf007, Rf008, Rf009, Rf010, Rf011, Rf012, Rf013, Rf014
Propósito:	Seleccionar las capas que se consideren necesarias para visualizarlas en el mapa de la Provincia de Loja, además manipular las capas e interactuar directamente con el mapa.
Descripción:	Mediante las opciones que ofrece el sistema el usuario puede activar o desactivar una nueva capa para visualizar en el mapa de la provincia de Loja; además mediante las herramientas que presenta el sistema se puede manipular la vista de las capas, es decir ampliar, reducir, girar; y algo muy importante se puede acceder directamente a la información de desastres naturales generados, ya que el usuario puede interactuar con el mapa.
Tipo:	Primario Real
Actor:	Usuario.
Precondición:	El usuario debe haber ingresado previamente al sistema El usuario debe estar en la Ventana de Visualización Gráfica El sistema debe estar conectado con la base de datos creada en MySQL



<p>Curso normal de eventos:</p>	<ol style="list-style-type: none"> 1. El usuario activa o desactiva las capas que quiere visualizar en el mapa de la Provincia de Loja 2. El sistema actualiza automáticamente el mapa con las capas que el usuario seleccionó. 3. El usuario visualiza la leyenda que identifica si la capa es un polígono, una línea o un punto; y además define el color de la capa 4. El usuario realiza acercamientos, alejamientos, giros y volver al estado inicial directamente en el mapa, mediante los controles que ofrece el sistema. 5. El sistema permite ver el mapa en el estado que el usuario solicita. 6. El usuario selecciona el botón “ARCHIVO SHAPE” en el sistema 7. El sistema presenta una ventana que contiene todas las opciones para actualizar, eliminar o cargar una nueva capa 8. El usuario selecciona una capa de la provincia o de un cantón. 9. El usuario activa la casilla <i>Nuevo Archivo</i> 10. El sistema activa automáticamente los campos: nombre, orden capa, color leyenda y directorio 11. El usuario ingresa los datos del nuevo archivo temporal y presiona el botón “ACEPTAR” 12. El sistema guarda los cambios en la capa y la muestra en la Ventana de Visualización Gráfica. 13. El usuario selecciona el botón “REFRESCAR” 14. El sistema vuelve a su estado original con los archivos base que contiene y que no pueden ser eliminados. 15. El Caso de Uso finaliza.
<p>Curso alterno de eventos:</p>	<p>A. Actualización de capas</p> <p>A9. El usuario activa la casilla <i>actualizar</i></p> <p>A10. El sistema activa automáticamente los campos: nombre, orden capa y color leyenda, que hasta ese momento no eran</p>

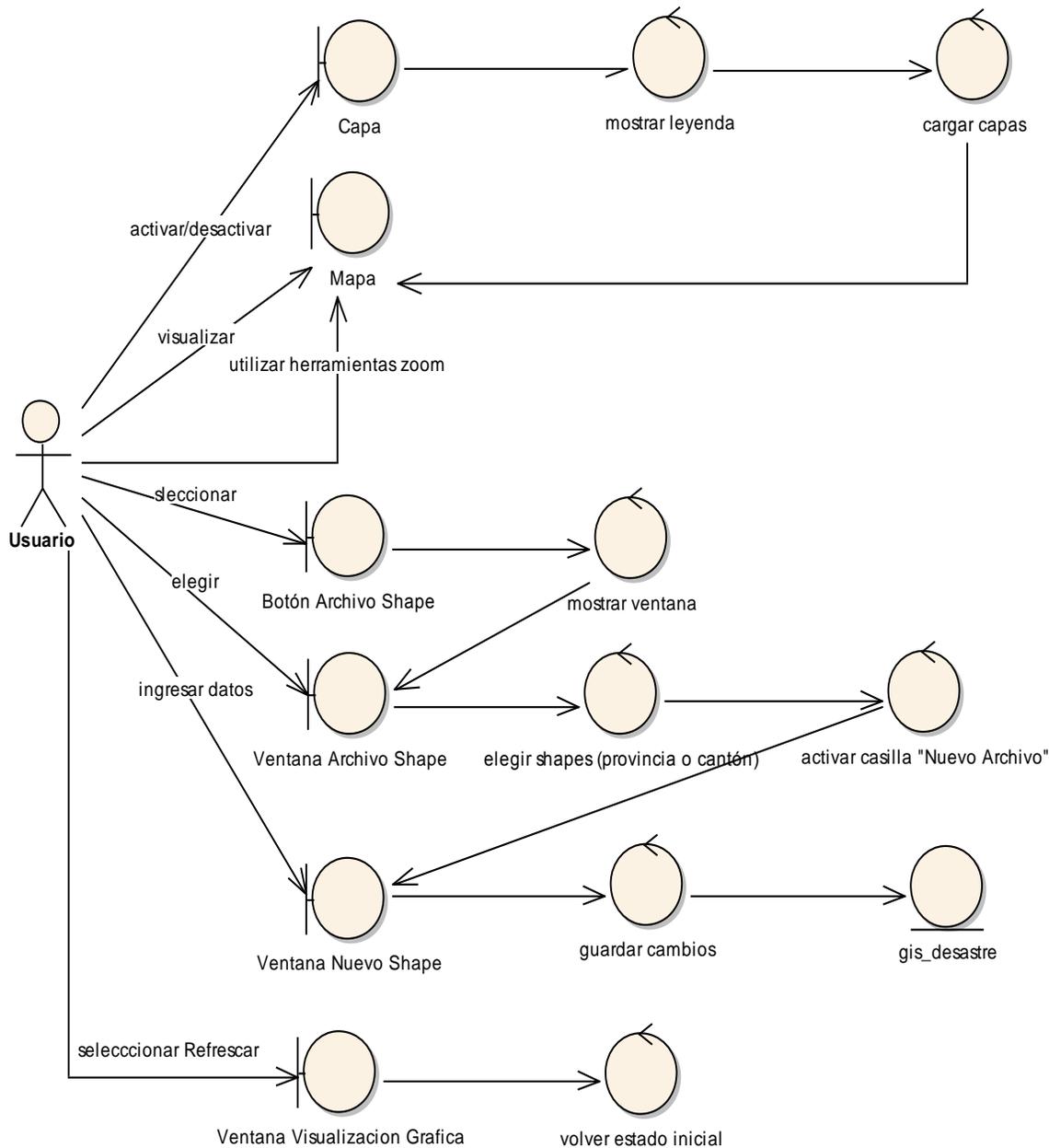


	<p>editables</p> <p>A11. El usuario digita el nuevo nombre de la capa y selecciona el orden en que quiere que aparezca la capa.</p> <p>A12. El usuario elige cambiar el color de la capa</p> <p>A13. El sistema presenta una paleta de colores donde se puede elegir el nuevo color de la capa</p> <p>A14. El usuario elige el nuevo color y presiona el botón “ACEPTAR” de la paleta de colores</p> <p>A15.- El usuario presiona el botón “ACEPTAR” de la Ventana Actualización</p> <p>A16. El Caso de Uso continúa en el paso 11 del curso normal de eventos</p> <p>B. Eliminación de Capas</p> <p>B9. El usuario activa la casilla <i>Eliminar</i></p> <p>B10. El sistema muestra un mensaje para confirmar si realmente desea eliminar la capa.</p> <p>B11. El usuario confirma la petición de borrado.</p> <p>B12. El sistema borra la capa de la base de datos (solo si no es de tipo base)</p> <p>B13. Si la capa es de tipo base, el sistema muestra un aviso de que no se puede eliminar este tipo de archivos.</p> <p>B14. El usuario confirma el aviso</p> <p>B15. El caso de uso finaliza.</p>
Post Condiciones:	<ul style="list-style-type: none"> ➤ Seleccionar las capas necesarias para mostrar en el sistema. ➤ Modificar los atributos (color, nombre, orden) de las capas base, mediante el uso de la paleta de colores. ➤ Aplicar los controles (acercamientos, alejamientos, giros) sobre el mapa base. ➤ Realizar cambios temporales (añadir, eliminar) en las capas

CUADRO 13: Caso de Uso # 3



✓ Diagrama de Robustez



Continúa en el Curso
Alternativo de Eventos

FIGURA 31: Diagrama de Robustez # 3

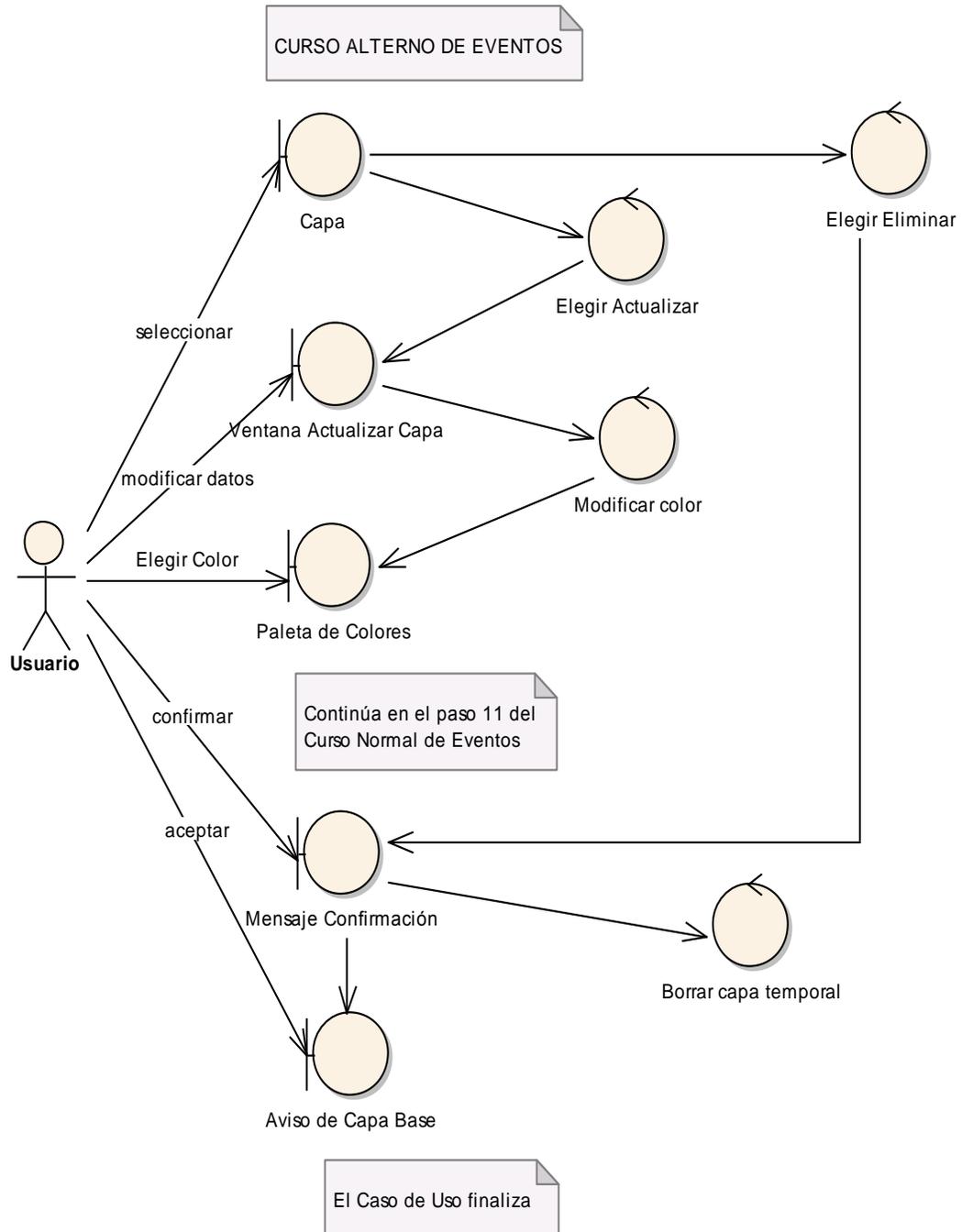


FIGURA 32: Diagrama de Robustez # 3 (Curso Alterno de Eventos)



✓ Diagrama de Secuencia

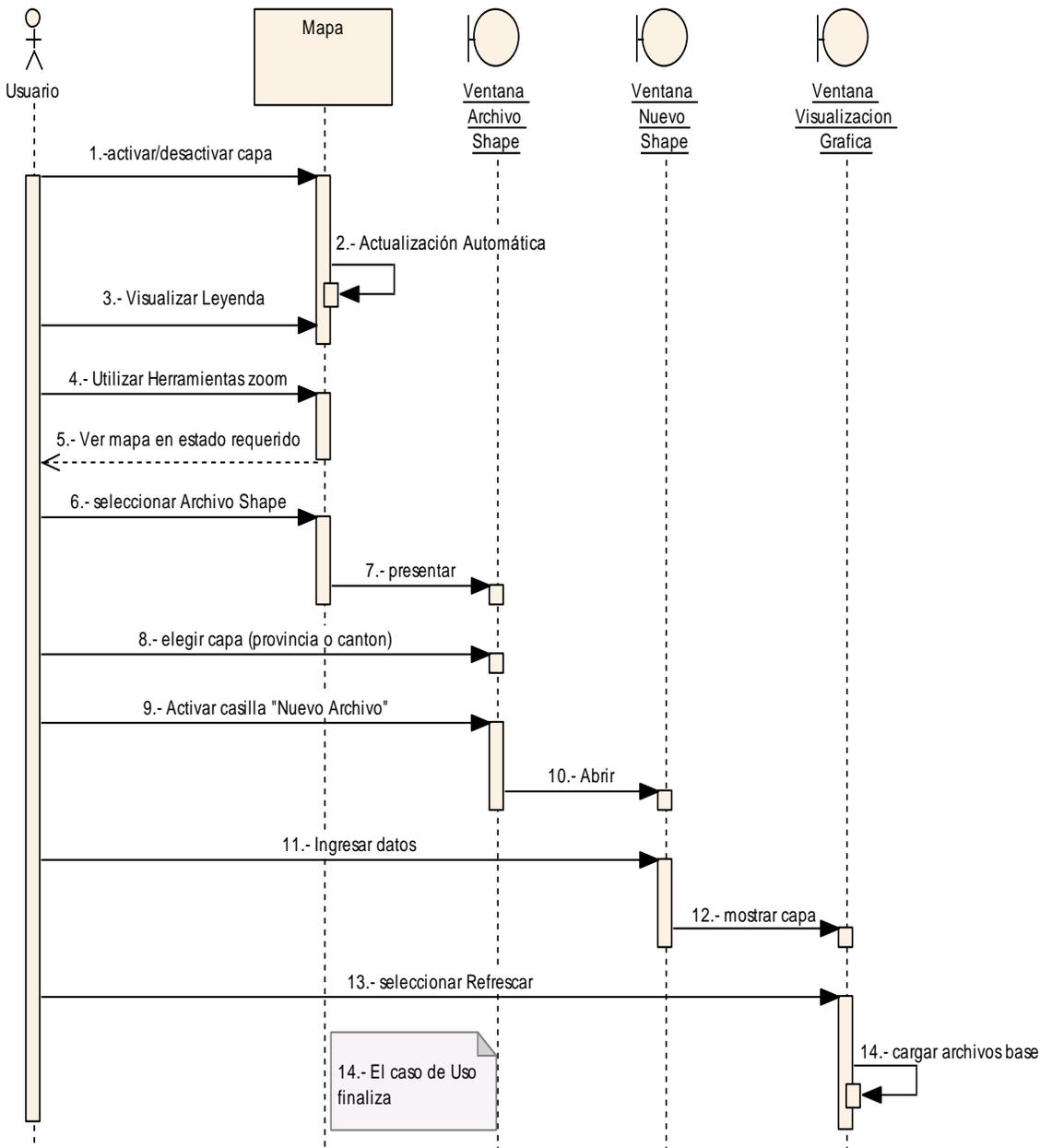


FIGURA 33: Diagrama de Secuencia # 3

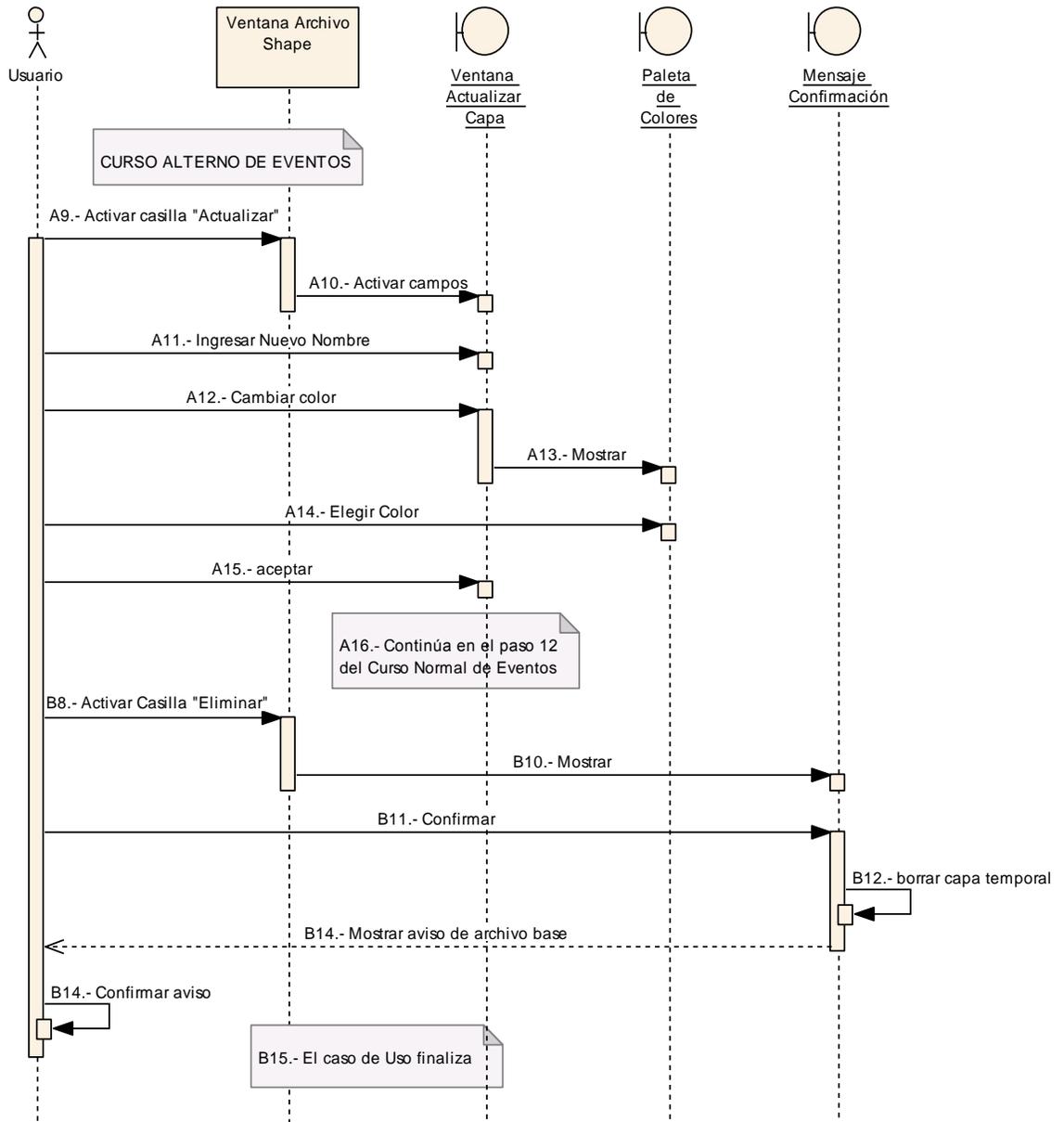


FIGURA 34: Diagrama de Secuencia # 3
(Curso Alternativo de Eventos)



7.4.2.4 Caso de Uso # 4: Ingresar información de nuevos desastres generados

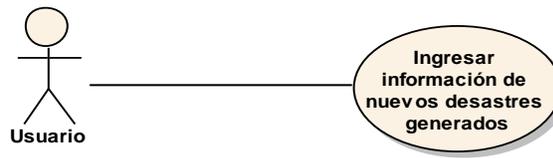


FIGURA 35: UC04

CASO DE USO :	Ingresar información de nuevos desastres generados
Identificador	UC04
Ref. Req.:	Rf015, Rf024, Rf025, Rf026, Rf027, Rf028
Propósito:	Ingresar un nuevo desastre al sistema con toda la información que le corresponde. Además diferenciar un cantón en el mapa, resaltándolo con un color diferente.
Descripción:	Mediante la ventana de Agregar Desastre que muestra el sistema, el usuario puede ingresar información de los nuevos desastres ocurridos, como es: localización, fecha, importancia, costo, coordenadas, tipo, parroquias afectadas, damnificados, muertos, heridos, imágenes, efectos y un informe general del desastre.
Tipo:	Primario Real.
Actor:	Usuario.
Precondición:	El usuario debe localizar el botón Agregar Desastre El sistema debe estar conectado con la base de datos
Curso normal de Eventos:	<ol style="list-style-type: none"> 1. El caso de uso se activa al momento que el usuario selecciona el botón “Agregar Desastre” 2. El sistema activa el botón “Ventana Desastre” y “Cantones” que hasta ese instante estaban desactivados; y carga solo las capas necesarias. 3. El usuario presiona el botón “Ventana Desastre” 4. El sistema muestra la Ventana para Agregar Desastre con



	<p>todos los campos vacíos y editables listos para guardar nueva información</p> <ol style="list-style-type: none"> 5. El usuario ingresa toda la información que posee del desastre generado y presiona el botón “GUARDAR”. 6. Si el usuario no ha ingresado las coordenadas, o estas no están dentro del área del cantón, el sistema dará un mensaje de error 7. El sistema guarda el nuevo desastre en la base de datos y lo muestra como punto en el mapa. 8. El caso de Uso finaliza
<p>Curso alternativo de Eventos:</p>	<p>A. Ingreso de un nuevo desastre desde el menú Edición.</p> <ol style="list-style-type: none"> A1. El usuario se ubica en el menú “Edición”, luego en “Datos Desastre Generado” y luego en “Insertar” A2. El sistema muestra la ventana de Agregar Desastre A3. El caso de uso continúa en el paso 4 del Curso Normal de Eventos. <p>B. Transformación de coordenadas</p> <ol style="list-style-type: none"> B5. El usuario selecciona la casilla “Geográficas” de la ventana de Agregar Desastre B6. El sistema presenta una ventana de transformación de coordenadas B7. El usuario ingresa las coordenadas de tipo geográficas (grados, minutos y segundos) para transformarlas a planas (metros) y presiona el botón “CALCULAR”. B8. El sistema muestra las coordenadas planas calculadas en metros y los resultados de los datos previos obtenidos. B9. El usuario presiona el botón “Descripción del Proceso” B10. El sistema muestra paso a paso el proceso que se siguió para el cálculo de las coordenadas B11. El usuario presiona el botón “ACEPTAR” B12. El sistema presenta las nuevas coordenadas calculadas en la ventana Agregar Desastre.



B13. El caso de uso continúa en el paso 5 del Curso Normal de Eventos

C. Ingreso de Imágenes

C5. El usuario presiona el botón “BUSCAR” para agregar imágenes al desastre

C6. El sistema muestra un directorio para buscar los archivos de tipo imagen

C7. El usuario elige las imágenes y presiona el botón “GUARDAR IMÁGENES”

C8. El sistema muestra una lista con las imágenes seleccionadas en la ventana Agregar Desastre

C9. El caso de uso continúa en el paso 5 del curso normal de eventos

D. Inserción de un Informe General

D5. El usuario presiona el botón “Redactar Informe”

D6.- El sistema muestra la ventana de Informe general con sus campos editables

D7. El usuario ingresa el informe general del desastre generado y presiona el botón “ACEPTAR”

D8. El sistema muestra el informe del desastre en la Ventana Agregar Desastre Generado

D9. El caso de uso continúa en el paso 5 del curso normal de eventos.

E. Resaltar Cantón

E3. El usuario presiona el botón “Cantones”.

E4. El sistema muestra la lista de todos los cantones de la Provincia de Loja

E5. El usuario selecciona un cantón de la lista.

E6. El sistema resalta el cantón en el mapa con un color diferente.

E7. El usuario puede observar y diferenciar el cantón.

E8. El caso de uso continúa en el paso 3 del curso Normal de Eventos



<p>Post</p> <p>Condiciones:</p>	<ul style="list-style-type: none"> ➤ Ingresar un nuevo desastre generado al sistema ➤ Diferenciar los cantones en el mapa ➤ Ingresar imágenes correspondientes a dicho desastre ➤ Ingresar un informe que describa a dicho desastre ➤ Transformar coordenadas si estas son de tipo geográficas
---	---

CUADRO 14: Caso de Uso # 4

✓ Diagrama de Robustez

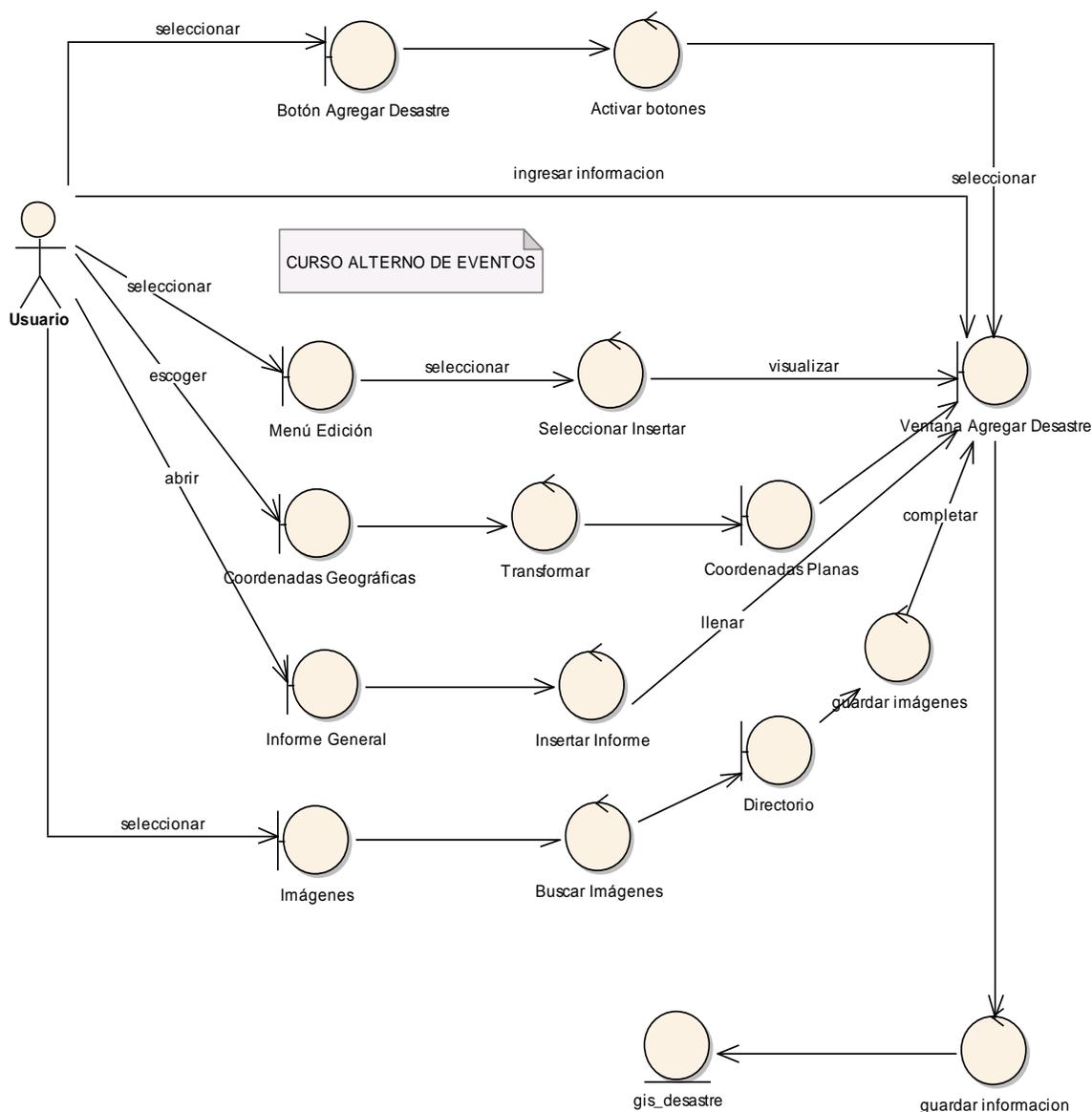


FIGURA 36: Diagrama de Robustez # 4



✓ Diagrama de Secuencia

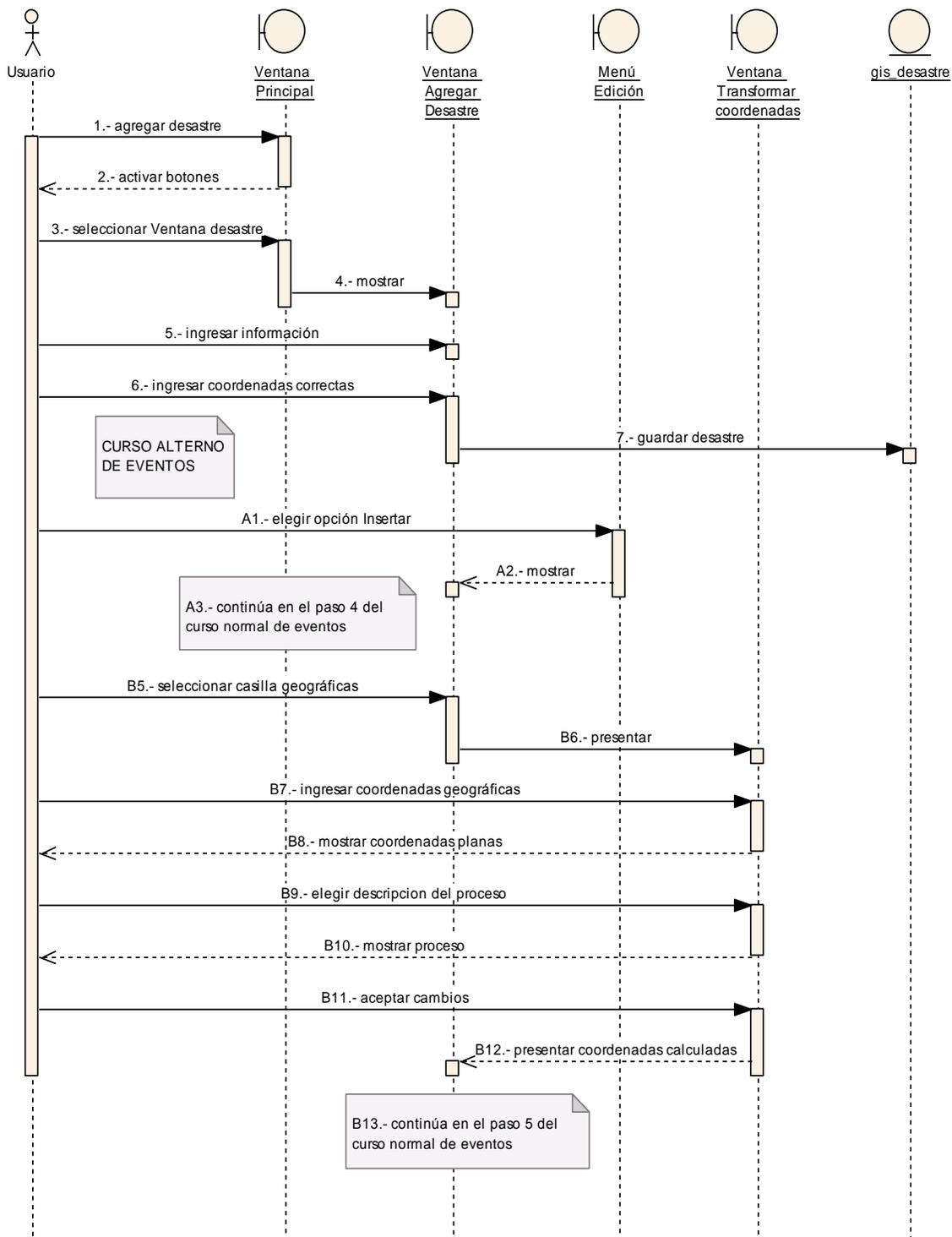
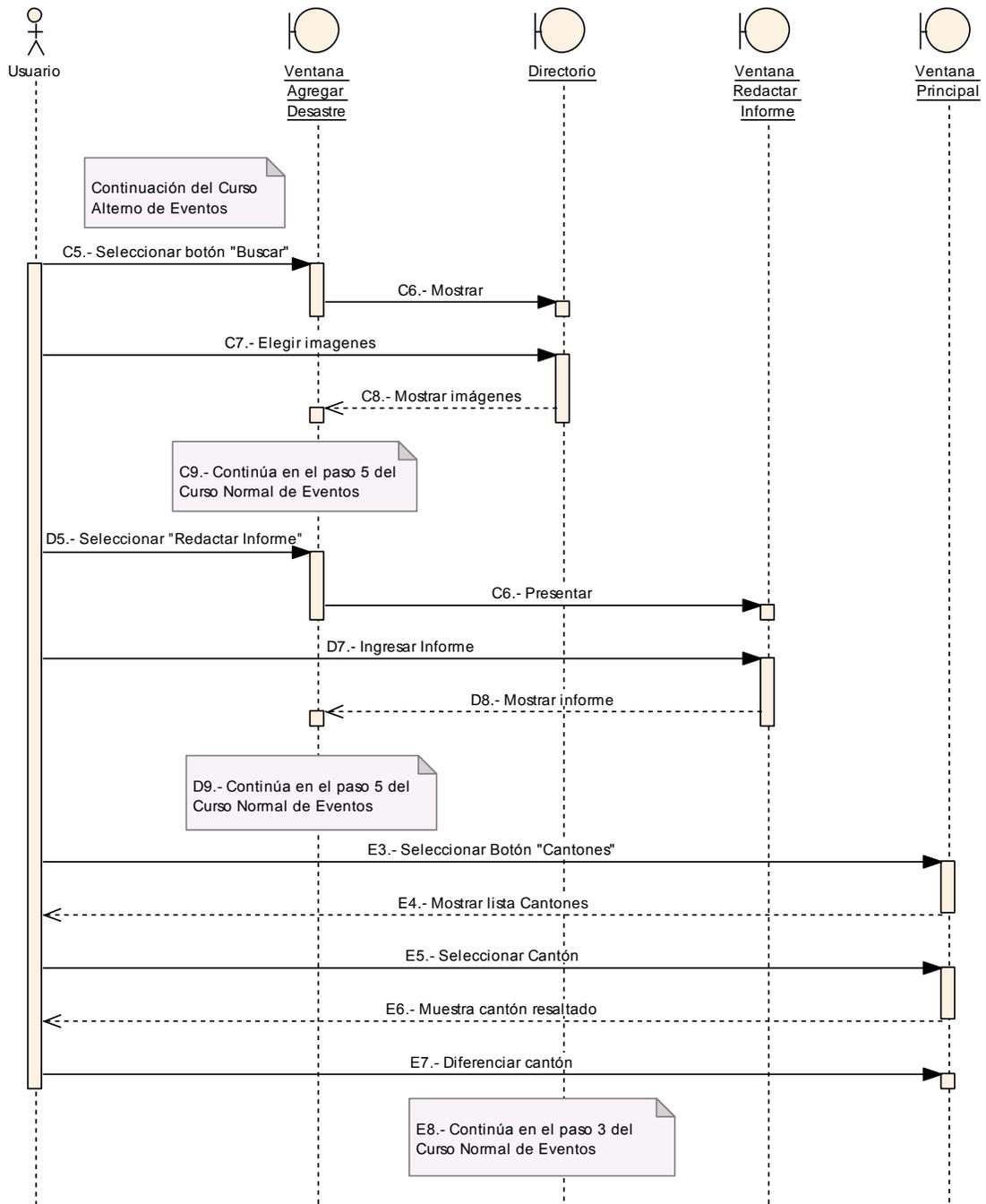


FIGURA 37: Diagrama de Secuencia # 4



**FIGURA 38: Diagrama de Secuencia # 4
(Curso Alternativo de Eventos)**



7.4.2.5 Caso de Uso # 5: Mostrar información de desastres generados.

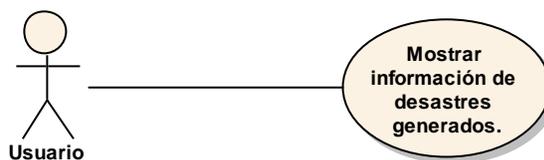


FIGURA 39: UC05

CASO DE USO :	Mostrar información de desastres generados.
Identificador:	UC05
Ref. Req.:	Rf016, Rf017, Rf019, Rf021
Propósito:	Acceder a la información completa de un desastre natural generado en la provincia de Loja; dicha información comprende: localización, fecha, importancia, costo, coordenadas, tipo, parroquias afectadas, damnificados, muertos, heridos, imágenes, efectos y un informe general del desastre.
Descripción:	El usuario puede interactuar directamente con el sistema para obtener información de un desastre generado que se encuentra registrado en la base de datos. Los desastres generados están ubicados como puntos en el mapa. Si el usuario selecciona uno de estos puntos se muestra la información completa del desastre generado.
Tipo:	Primario Real
Actor:	Usuario.
Precondición	El usuario debe ubicarse en el menú “HERRAMIENTAS” del sistema. El usuario debe saber los ID de cada uno de los desastres. El sistema debe estar conectado con la base de datos creada para el efecto en MySQL.



<p>Curso normal de eventos:</p>	<ol style="list-style-type: none"> 1. El usuario activa la casilla “BUSQUEDA MAPA” que se encuentra en el menú “HERRAMIENTAS” 2. El sistema prepara los puntos que se encuentran registrados en el mapa como desastres naturales para que el usuario pueda interactuar. 3. El usuario hace un clic sobre uno de los puntos marcados como desastres en el mapa. 4. El sistema busca la información del desastre seleccionado (coordenadas, damnificados, pérdidas, muertos, heridos, ubicación, efectos) en la base de datos. 5. El sistema muestra la Ventana de Desastre Generado 6. El usuario visualiza toda la información del desastre natural seleccionado 7. El caso de uso finaliza.
<p>Curso alternativo de eventos:</p>	<p>A. Búsqueda mediante ID</p> <ol style="list-style-type: none"> A1. El usuario se ubica en el menú “Edición”, luego en “Datos Desastre Generado” y luego en “Buscar” A2. El sistema muestra una ventana de búsqueda mediante el ID del desastre. A3. El usuario ingresa el ID del desastre generado A4. El caso de uso continúa en el paso 4 del Curso Normal de Eventos. <p>B. Visualización de imágenes</p> <ol style="list-style-type: none"> B5. El usuario visualiza las imágenes que le corresponden a ese desastre en el visor de imágenes de la misma ventana. B6. El usuario presiona el botón “Siguiente” para visualizar la siguiente imagen B7. El usuario selecciona el botón “GALERÍA” B8. El sistema presenta una ventana con toda la galería de imágenes del desastre. B9. El caso de uso continúa en el paso 5 del curso normal de eventos <p>C. Visualización de Informe General del desastre</p>



	<p>generado</p> <p>C5. El usuario presiona el botón “INFORME GENERAL”</p> <p>C6. El sistema presenta un informe general que contiene un resumen del desastre generado.</p> <p>C7. El caso de uso continúa en el paso 5 del curso normal de eventos.</p>
<p>Post</p> <p>Condiciones:</p>	<ul style="list-style-type: none"> ➤ Mostrar la información completa del desastre natural generado en la provincia ➤ Mostrar las imágenes reales del desastre natural ➤ Mostrar un informe general del desastre generado.

CUADRO 15: Caso de Uso # 4



✓ **Diagrama de Robustez**

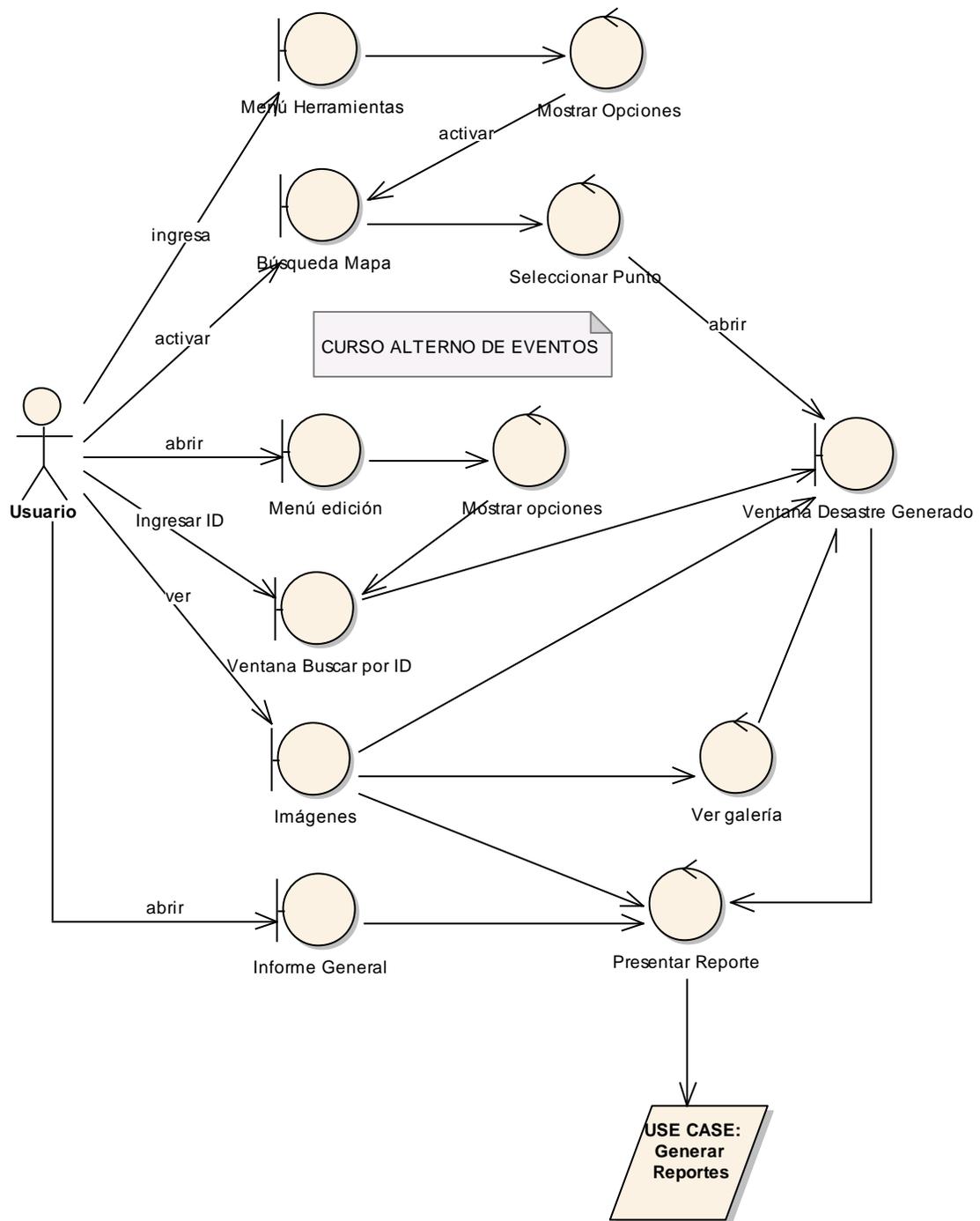


FIGURA 40: Diagrama de Robustez # 5



✓ Diagrama de Secuencia

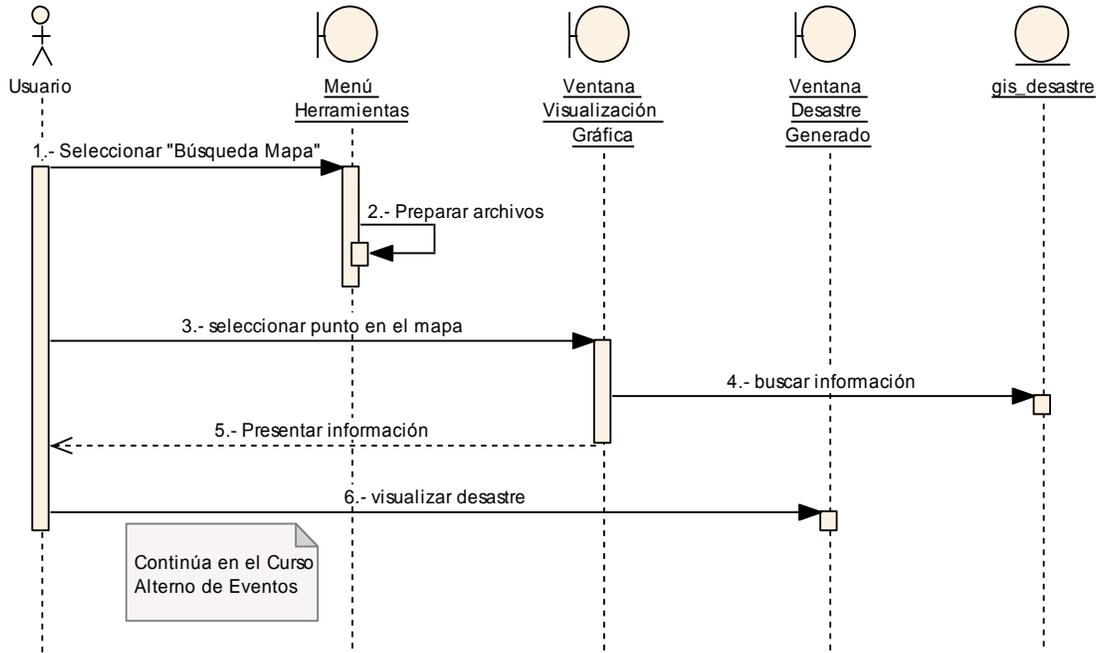
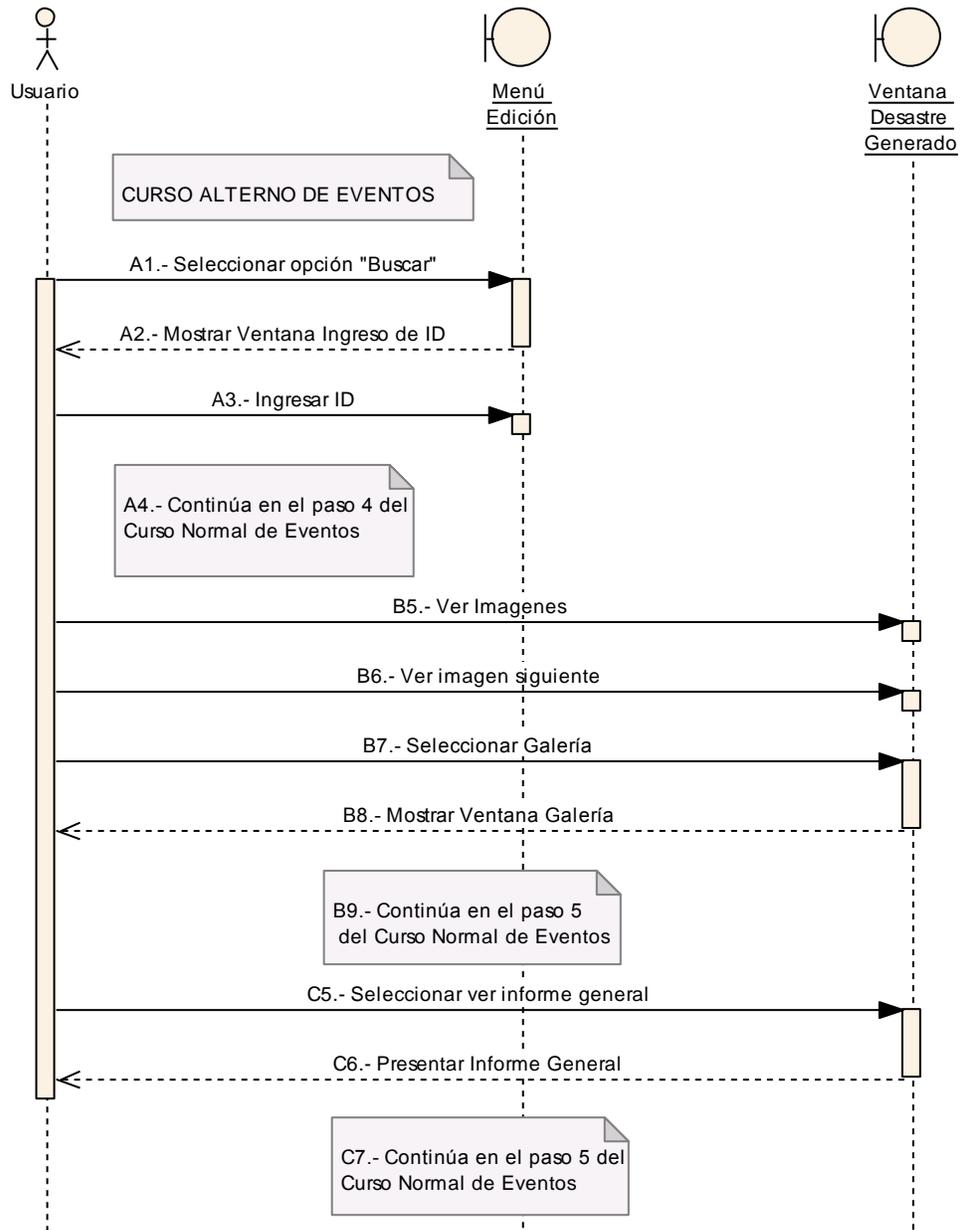


FIGURA 41: Diagrama de Secuencia # 5



**FIGURA 42: Diagrama de Secuencia # 5
(Curso Alterno de Eventos)**



7.4.2.6 Caso de Uso # 6: Actualizar y Eliminar la información existente de desastres generados.

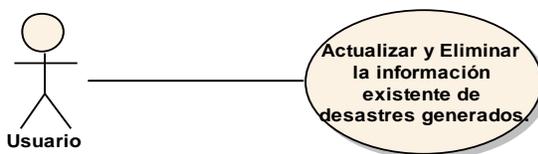


FIGURA 43: UC06

CASO DE USO :	Actualizar y Eliminar la información existente de desastres generados.
Identificador:	UC06
Ref. Req.:	Rf023, Rf025, Rf026, Rf027, Rf028
Propósito:	Modificar la información de los desastres naturales generados que se encuentran registrados en el sistema y eliminar un desastre natural generado por completo de la base de datos del sistema.
Descripción:	Mediante las opciones que ofrece el sistema, el usuario puede manipular la información de los desastres generados, es decir puede elegir si desea actualizar o eliminar los datos de un desastre generado.
Tipo:	Primario Real.
Actor:	Usuario.
Precondición:	El usuario debe haber accedido a la información de un desastre generado. El usuario debe conocer el ID de los desastres registrados El sistema debe estar conectado con la base de datos creada para el efecto en MySQL.
Curso normal de eventos:	<ol style="list-style-type: none"> 1. El usuario selecciona el botón "ACTUALIZAR" 2. El sistema muestra una ventana con todos los campos editables del desastre generado. 3. El usuario modifica las coordenadas y verifica que estén dentro del área del cantón 4. El usuario modifica el resto de información del desastre generado, teniendo en cuenta los campos que aceptan solo



	<p>valores enteros.</p> <p>5. El usuario presiona el botón “ACEPTAR”</p> <p>6. El sistema guarda los cambios y actualiza el desastre.</p> <p>7. El caso de uso finaliza</p>
Curso alternativo de eventos:	<p>i. Actualización mediante ID</p> <p>A2. El usuario se ubica en el menú “Edición”, luego en “Datos Desastre Generado” y luego en “Actualizar”</p> <p>A3. El sistema muestra una ventana de actualización mediante el ID del desastre.</p> <p>A4. El usuario ingresa el ID del desastre generado</p> <p>A5. El caso de uso continúa en el paso 2 del Curso Normal de Eventos.</p> <p>ii. Transformación de coordenadas</p> <p>B4. El usuario selecciona la casilla “Geográficas” de la ventana de Actualización</p> <p>B5. El sistema presenta una ventana de transformación de coordenadas</p> <p>B6. El usuario ingresa las coordenadas de tipo geográficas (grados, minutos y segundos) para transformarlas a planas (metros) y presiona el botón “CALCULAR”.</p> <p>B7. El sistema muestra las coordenadas planas calculadas en metros y los resultados de los datos previos obtenidos en la misma ventana.</p> <p>B8. El usuario presiona el botón “Descripción del Proceso”</p> <p>B9. El sistema muestra paso a paso el proceso que se siguió para el cálculo de coordenadas</p> <p>B10. El usuario presiona el botón “ACEPTAR”</p> <p>B11. El sistema presenta las nuevas coordenadas calculadas en la ventana de actualización.</p> <p>B12. El caso de uso continúa en el paso 4 del Curso Normal de Eventos</p> <p>iii. Cambiar Imágenes</p> <p>C4. El usuario presiona el botón “BUSCAR” para agregar</p>



	<p>nuevas imágenes al desastre</p> <p>C5. El sistema muestra un directorio para buscar los archivos de tipo imagen</p> <p>C6. El usuario elige las imágenes y presiona el botón “GUARDAR IMÁGENES”</p> <p>C7. El sistema muestra una lista con las imágenes seleccionadas en la ventana de actualización</p> <p>C8. El caso de uso continúa en el paso 4 del curso normal de eventos</p> <p>iv. Redactar Informe</p> <p>D4. El usuario presiona el botón “Redactar Informe”</p> <p>D5.- El sistema muestra una ventana con el texto del informe editable</p> <p>D6. El usuario modifica el informe general del desastre generado y presiona el botón “ACEPTAR”</p> <p>D7. El sistema presenta el nuevo informe en la Ventana Actualización de Desastre Generado</p> <p>D8. El caso de uso continúa en el paso 4 del curso normal de eventos</p> <p>v. Eliminar un desastre generado</p> <p>E1. El usuario selecciona el botón “BORRAR”</p> <p>E2. El sistema muestra un mensaje de confirmación, de que si desea borrar o no el desastre generado de la base de datos.</p> <p>E3. El usuario confirma la petición al sistema</p> <p>E4. El sistema borra por completo el desastre generado de la base de datos</p> <p>E5. El caso de uso finaliza.</p>
<p>Post Condiciones:</p>	<ul style="list-style-type: none"> ➤ Modificar la información mal ingresada de un desastre generado ➤ Agregar imágenes al desastre ➤ Cambiar el informe general del desastre generado ➤ Eliminar un desastre generado de la base de datos. ➤ Actualizar un desastre natural

CUADRO 16: Caso de Uso # 6



✓ Diagrama de Robustez

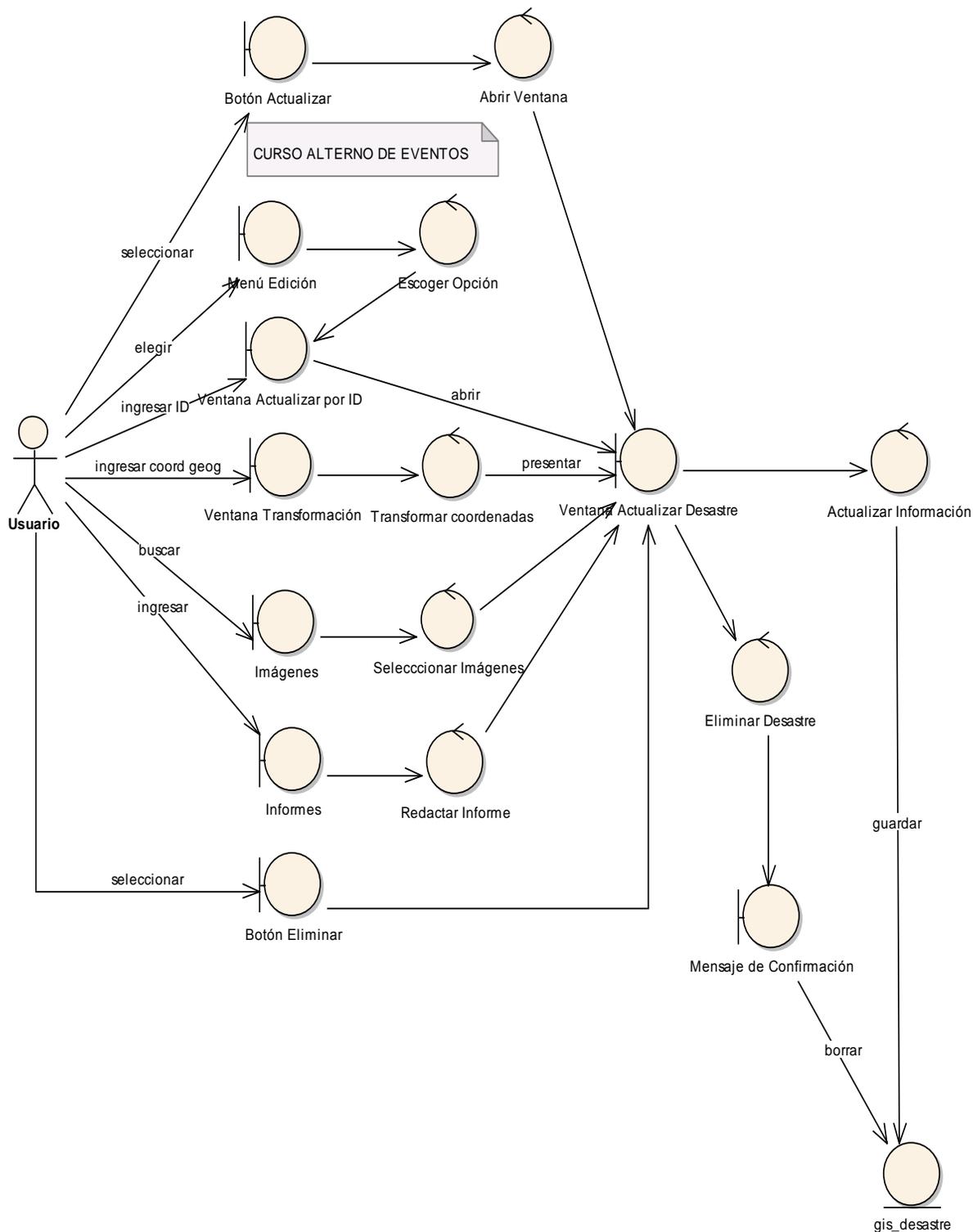


FIGURA 44: Diagrama de Robustez # 6



✓ Diagrama de Secuencia

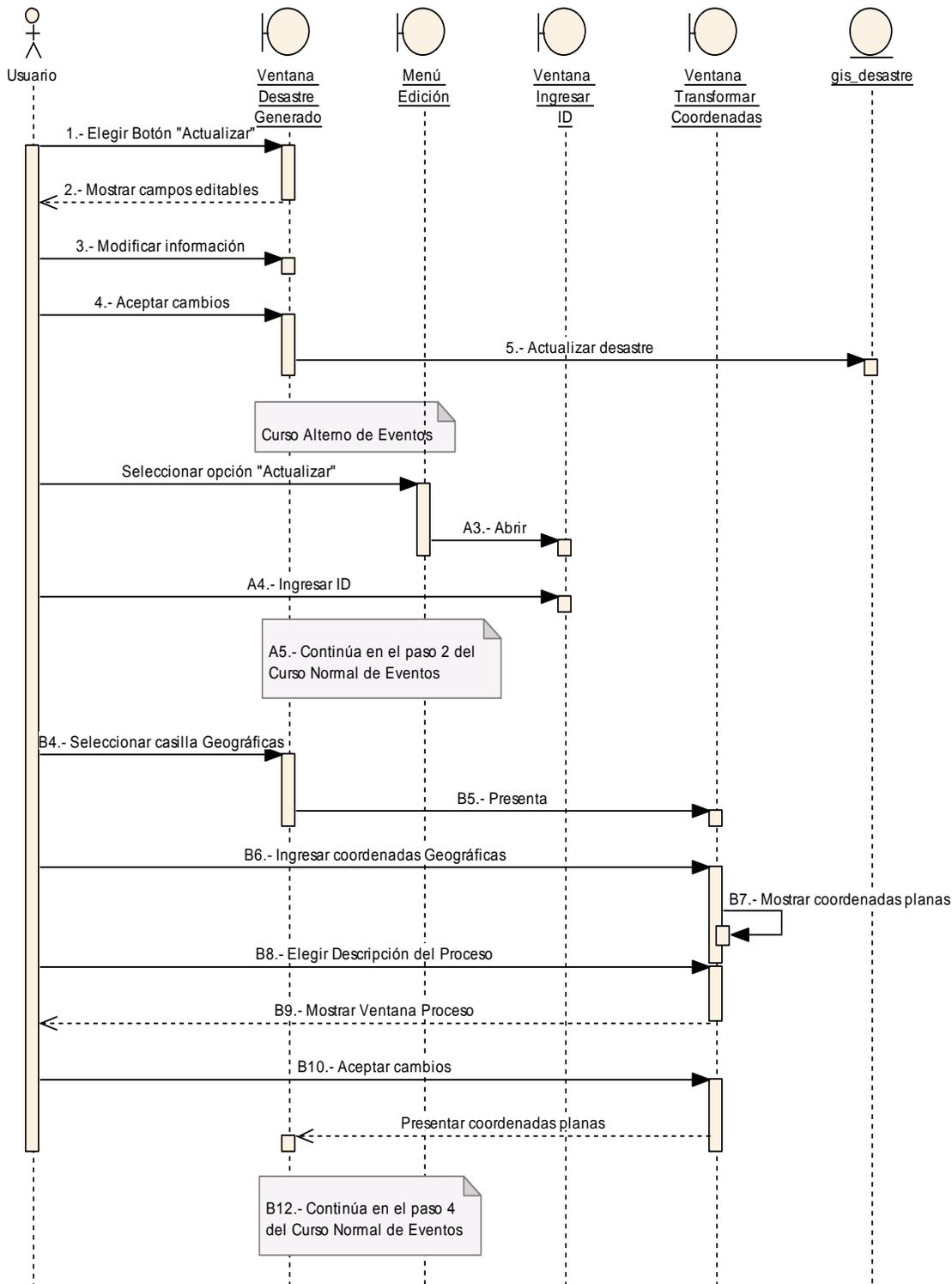


FIGURA 45: Diagrama de Secuencia # 6

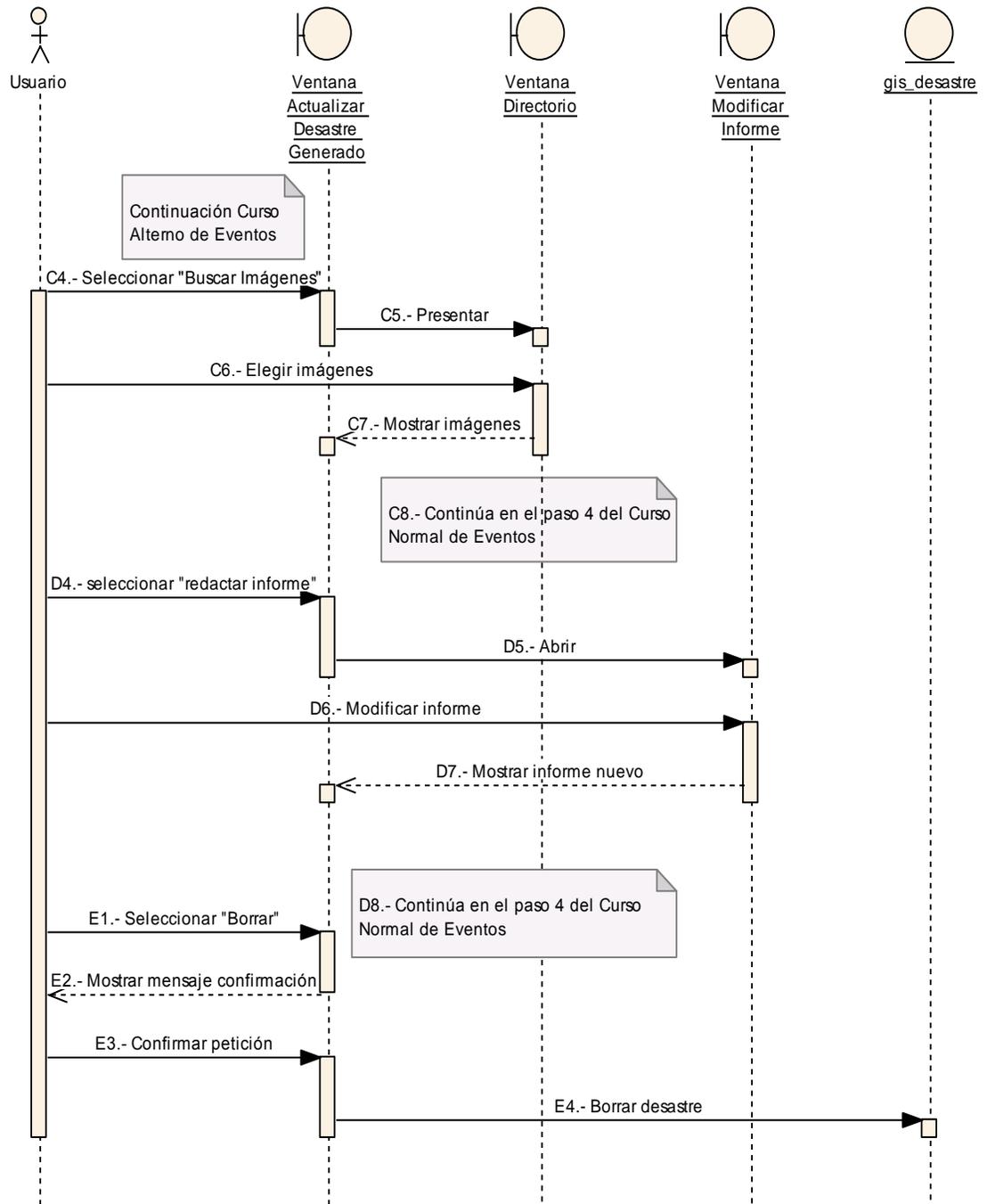


FIGURA 46: Diagrama de Secuencia # 6
(Curso Alternativo de Eventos)

7.4.2.7 Caso de Uso # 7: Filtrar información específica de desastres generados

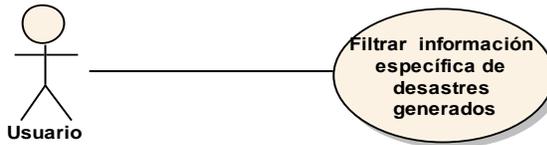


FIGURA 47: UC07

CASO DE USO :	Filtrar información específica de desastres generados
Identificador	UC07
Ref. Req.:	Rf029, Rf031, Rf032, Rf033, Rf034
Propósito:	Permitir al usuario que pueda realizar consultas en el sistema de los desastres naturales generados que tiene registrados, mediante distintas opciones consideradas.
Descripción	El usuario abre la ventana de búsquedas que contiene todas las opciones de filtrado de información de desastres generados. Mediante esta ventana el usuario puede realizar búsquedas, ya sea por provincia, cantón, tipo de desastre, fecha de ocurrencia; proporcionando así una búsqueda específica.
Tipo:	Primario Real
Actor:	Usuario.
Precondición:	El usuario se debe ubicar el botón Búsquedas El sistema debe estar conectado con la base de datos creada para el efecto en MySQL.
Curso normal de eventos:	<ol style="list-style-type: none"> 1. El caso de uso se activa al momento de hacer clic en el botón "BUSQUEDAS" 2. El sistema presenta una ventana de filtrado de datos con varias opciones para filtrar la información: por cantones, por tipo de desastre, por fecha de ocurrencia. 3. El usuario escoge una de estas opciones y presiona el botón "FILTRAR" 4. El sistema busca la selección requerida en la base de datos



	<p>5. El sistema muestra una tabla con los resultados obtenidos</p> <p>6. El usuario activa la casilla “Ventana Datos” y selecciona uno de los desastres</p> <p>7. El sistema muestra una ventana con la información completa del desastre generado</p> <p>8. El caso de uso finaliza.</p>
<p>Curso alterno de eventos:</p>	<p>1. Visualización del mapa</p> <p>A6. El usuario activa una de las opciones “Mapa Provincia” si quiere visualizar los puntos en el mapa de la provincia, o “Mapa Cantón” para ver los puntos en el mapa del cantón</p> <p>A7. El usuario selecciona el botón “VISUALIZAR MAPA”</p> <p>A8. El sistema muestra el mapa con todos los puntos filtrados y marcados con un triángulo</p> <p>A9. El usuario se ubica en el menú “HERRAMIENTAS” y selecciona la opción “Dibujar Puntos”</p> <p>A10. El sistema presenta los puntos filtrados en una lista</p> <p>A11. El usuario selecciona uno de los puntos</p> <p>A12. El sistema dibuja el punto como círculo y le asigna un color para que el usuario pueda distinguirlo</p> <p>A13. El usuario presiona el botón registro</p> <p>A14. El sistema presenta la información completa del desastre generado</p> <p>A15. El usuario selecciona el botón “Tabla Filtrado”</p> <p>A16. El sistema muestra la tabla con el resultado de la última consulta realizada.</p> <p>A17. El caso de uso continúa en el paso 6 del curso normal de eventos</p>
<p>Post Condiciones</p>	<ul style="list-style-type: none"> ➤ Mostrar las opciones de filtrado de desastres naturales ➤ Mostrar una tabla con las consultas que se realizan ➤ Visualizar los puntos filtrados en el mapa, ya sea de la provincia o del cantón

CUADRO 17: Caso de Uso # 7

✓ **Diagrama de Robustez**

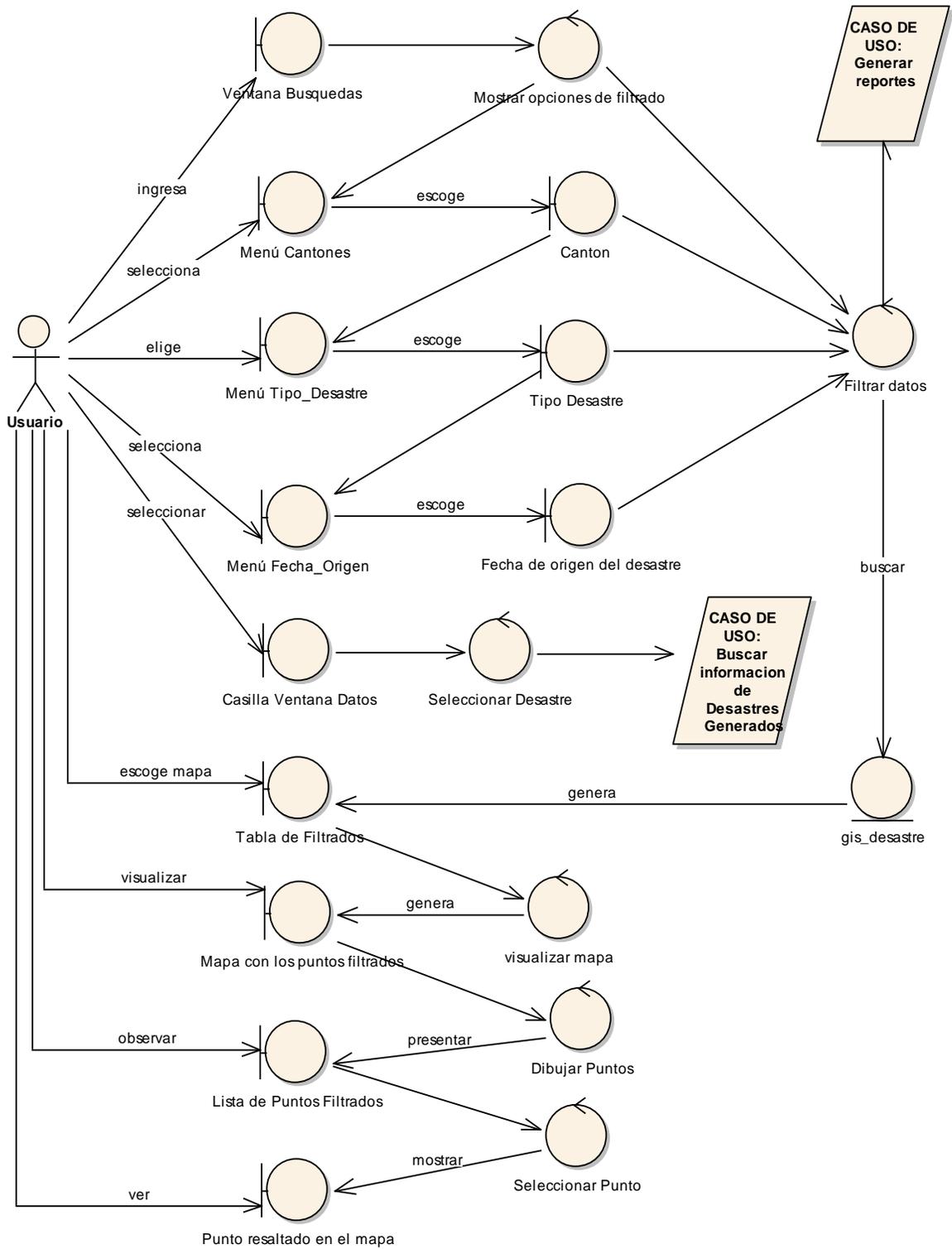


FIGURA 48: Diagrama de Robustez # 7

✓ **Diagrama de Secuencia**

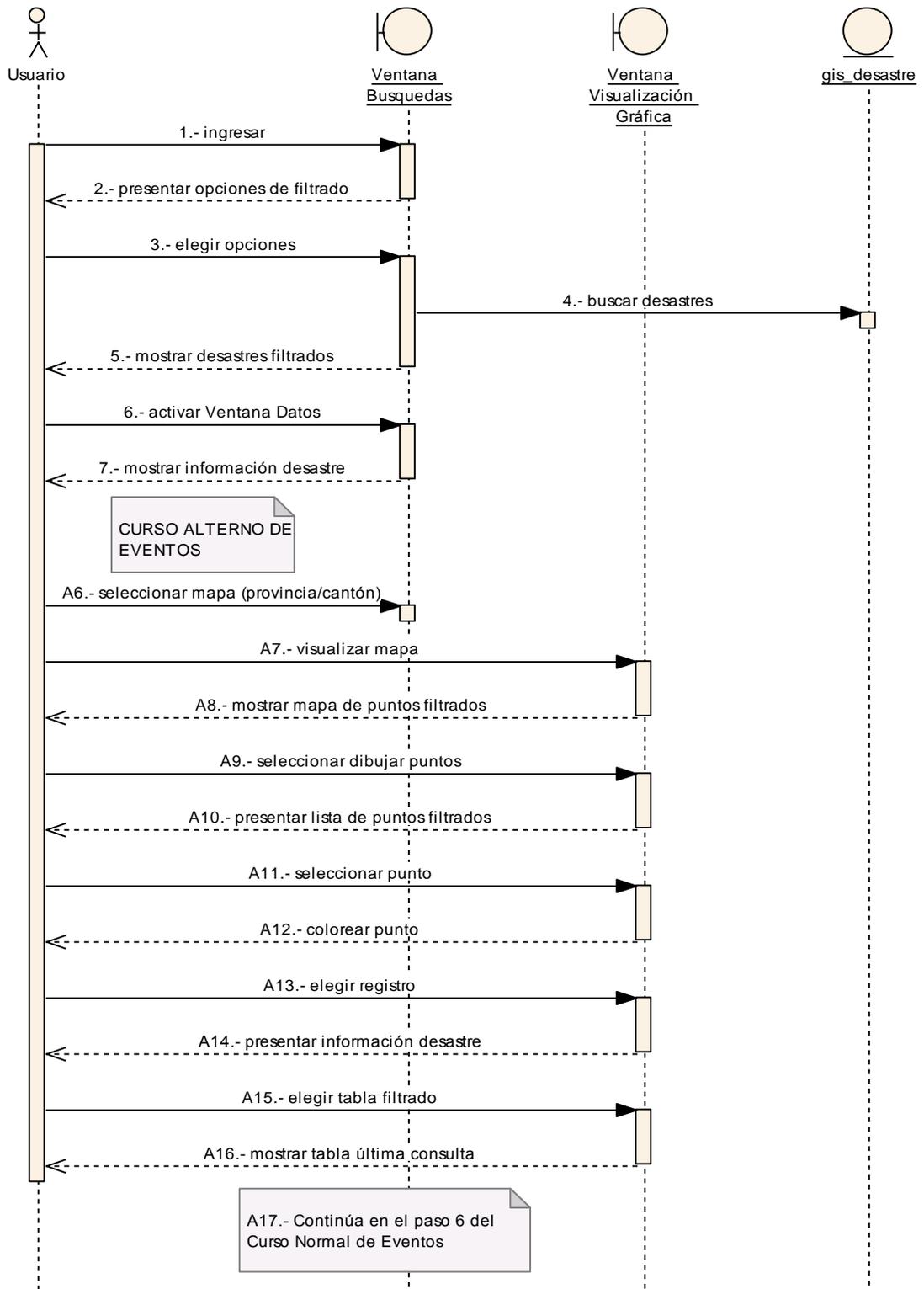


FIGURA 49: Diagrama de Secuencia # 7

7.4.2.8 Caso de Uso # 8: Generar reportes

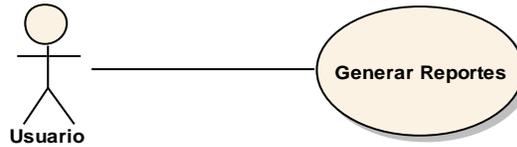


FIGURA 50: UC08

CASO DE USO :	Generar reportes
Identificador	UC08
Ref. Req.:	Rf020, Rf022, Rf030
Propósito:	Presentar un reporte imprimible para cada desastre natural generado en la provincia de Loja, y para las consultas que se realicen en el sistema sobre desastres
Descripción:	El usuario puede generar un reporte con solo presionar el botón “REPORTE” que se encuentra tanto en la ventana del desastre generado, como en la ventana de búsquedas.
Tipo:	Primario Real
Actor:	Usuario.
Precondición:	El usuario debe haber accedido a la información de un desastre generado. El usuario debe ubicar el botón “REPORTE” que se encuentra en la ventana de desastre generado El sistema debe estar conectado con la base de datos creada para el efecto en MySQL.
Curso normal de Eventos:	<ol style="list-style-type: none"> 1. El caso de uso se activa al momento que el usuario presiona el botón “REPORTE” de la ventana Desastre Generado. 2. El sistema muestra un reporte con todos los datos generales de ese desastre, obtenidos de la base de datos. 3. El usuario puede seleccionar el zoom del reporte (%) 4. El usuario puede guardar el reporte en una dirección en la extensión que lo requiera (pdf, excel).



	<p>5. Si el usuario lo requiere, puede mandar a imprimir directamente el reporte</p> <p>6. El caso de uso finaliza</p>
Curso alternativo de Eventos:	<p>A. Generación del reporte de Búsquedas</p> <p>A1. El usuario accede a la ventana de filtrado de desastres</p> <p>A2. El usuario selecciona los desastres de los que quiere generar un reporte</p> <p>A3. El sistema prepara los desastres para generar el reporte</p> <p>A4. El usuario presiona el botón “Crear Reporte”</p> <p>A5. El sistema muestra un reporte imprimible con los datos de los desastres que se enviaron a buscar.</p> <p>A6. Continúa en el paso 3 del Curso Normal de Eventos</p> <p>B. Reporte Imagen</p> <p>B3. El usuario presiona el botón Reporte de la imagen seleccionada</p> <p>B4. El sistema muestra un reporte con esa imagen y los datos de ese desastre generado.</p> <p>B5. El caso de uso continúa en el paso 3 del curso normal de eventos</p> <p>C. Reporte Informe General</p> <p>C3. El usuario ingresa al informe general del desastre generado.</p> <p>C4. El usuario selecciona el botón “Reporte” dentro de la ventana de Informe General.</p> <p>C5. El sistema muestra un reporte con el informe general del desastre generado.</p> <p>C6. Continúa en el paso 3 del curso normal de eventos.</p>
Post Condiciones:	<ul style="list-style-type: none"> ➤ Presentar un reporte de un desastre generado ➤ Presentar reportes para las imágenes ➤ Presentar reportes para el Informe general del desastre ➤ Presentar reportes para las búsquedas o filtrados.

CUADRO 18: Caso de Uso # 8

✓ **Diagrama de Robustez**

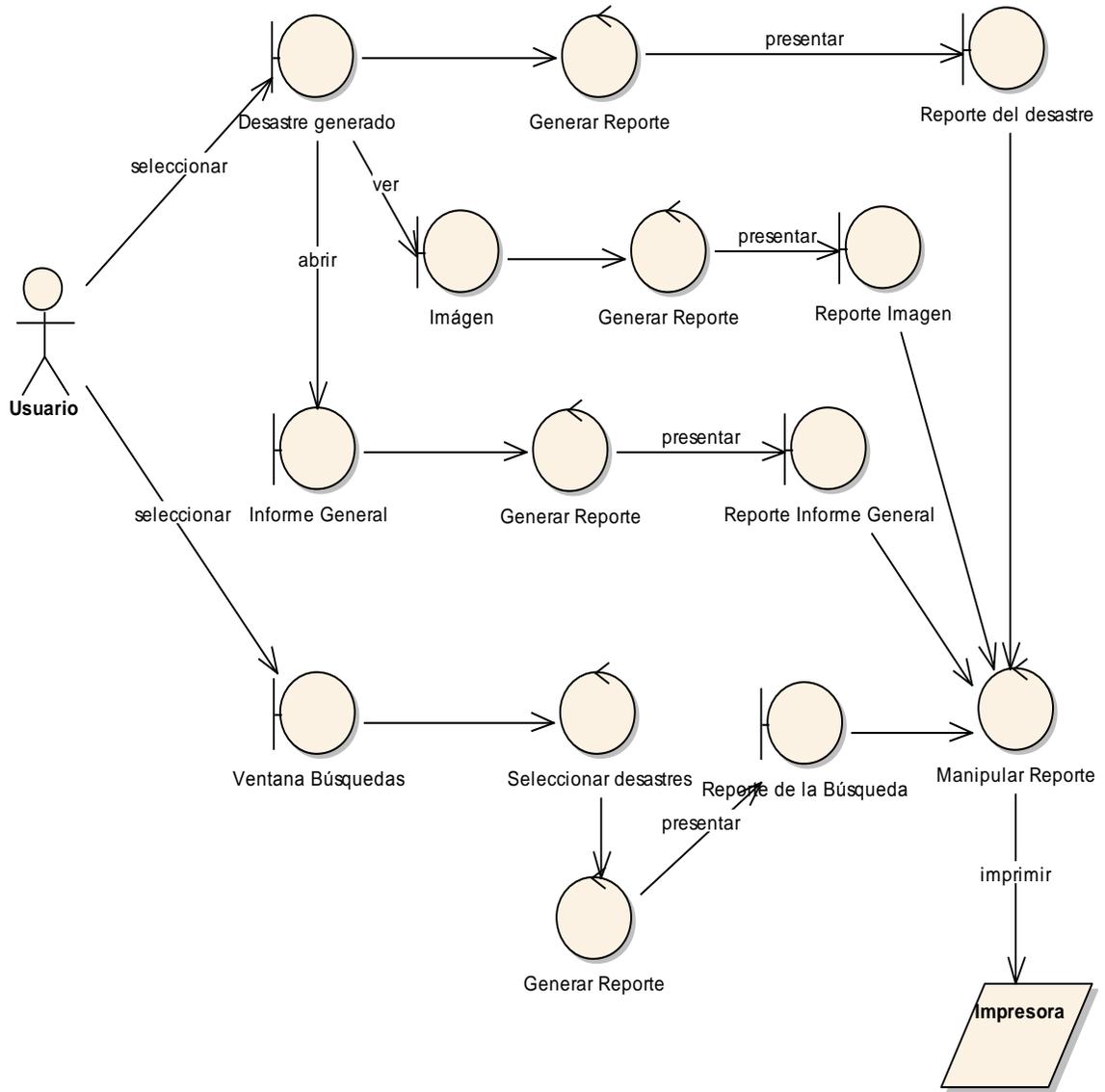


FIGURA 51: Diagrama de Robustez # 8

✓ **Diagrama de Secuencia**

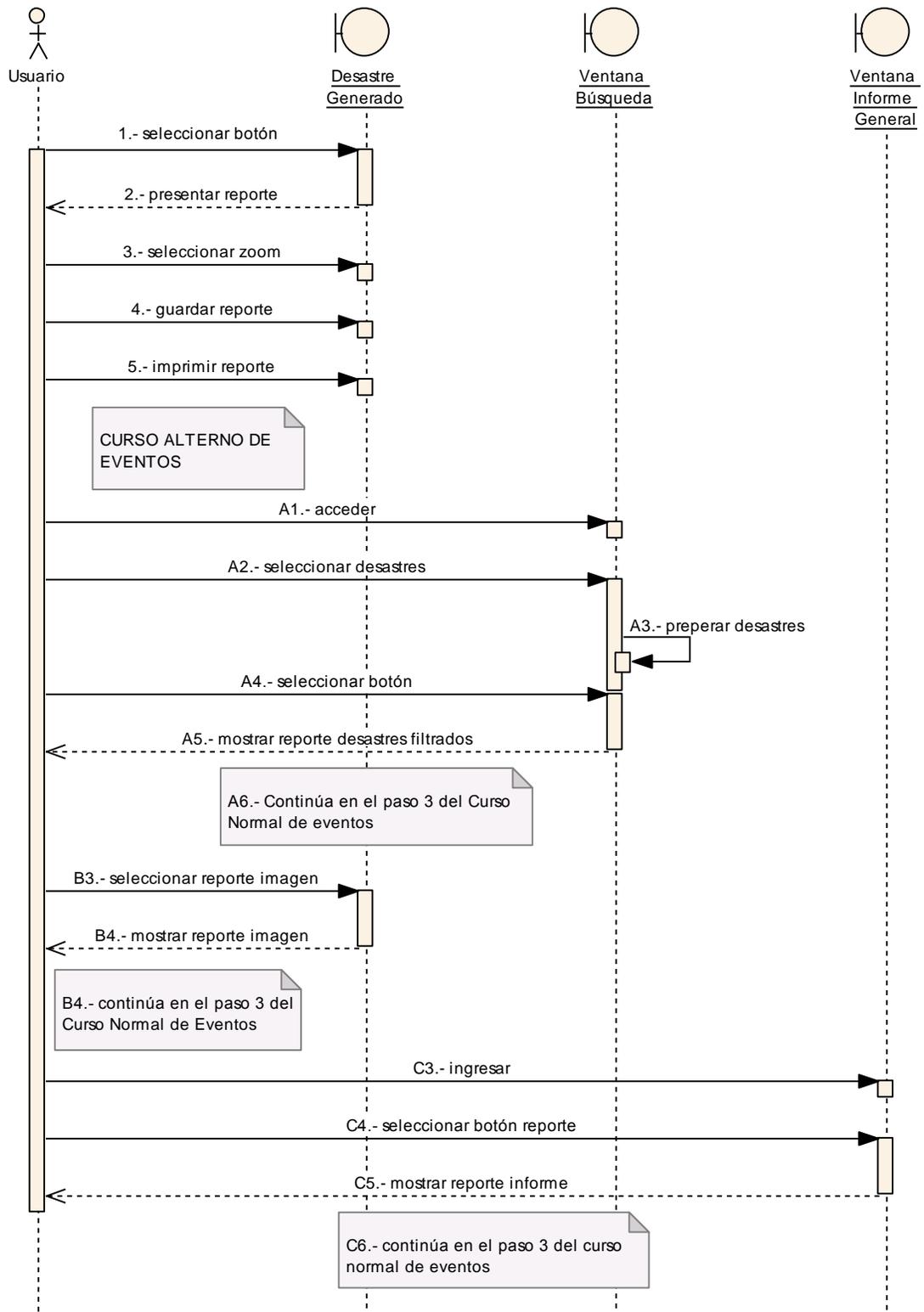


FIGURA 52: Diagrama de Secuencia # 8



7.4.2.9 Caso de Uso # 9: Consultar información de la provincia y obtener ayuda del SIG

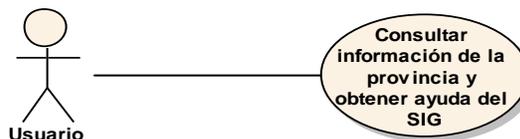


FIGURA 53: UC09

CASO DE USO :	Consultar información de la provincia y obtener ayuda del SIG
Identificador:	UC09
Ref. Req.:	Rf035, Rf036, Rf037
Propósito:	Obtener información general de la provincia y de cada uno de sus cantones, así como una breve descripción de lo que es un SIG. Además mostrar todo el manual del usuario para el manejo adecuado del sistema
Descripción:	El usuario puede consultar en el sistema la información más relevante (historia, costumbres, ubicación, etc.) de la provincia de Loja, así como de los cantones que le pertenecen y puede observar una descripción del SIG. Así mismo mediante la opción “AYUDA” puede tener un soporte para el manejo adecuado del sistema.
Tipo:	Primario Real.
Actor:	Usuario.
Precondición:	El usuario debe estar ubicado en la pestaña “Información General” en el sistema El sistema debe estar conectado con la base de datos creada para el efecto en MySQL.
Curso normal de Eventos:	<ol style="list-style-type: none"> 1. El usuario selecciona la pestaña “Información General” 2. El sistema muestra 3 opciones “Provincia”, “G.I.S” y “Manejo GIS Ayuda” 3. El usuario selecciona la opción “Provincia” 4. El sistema muestra una ventana con la información de la



	<p>Provincia y sus cantones.</p> <p>5. El usuario puede visualizar y si lo requiere copiar la información que considere importante</p> <p>6. El caso de uso termina.</p>
Curso alternativo de Eventos:	<p>A. Descripción GIS</p> <p>A3. El usuario selecciona la opción “GIS”</p> <p>A4. El sistema muestra una ventana con la descripción del GIS</p> <p>A5. El caso de uso continúa en el paso 5 del Curso Normal de Eventos</p> <p>B. Ayuda del Sistema</p> <p>B3. El usuario selecciona la opción “Ayuda GIS”</p> <p>B4. El sistema muestra una ventana con la ayuda para el correcto manejo del sistema</p> <p>B5. El caso de uso continúa en el paso 5 del Curso Normal de Eventos.</p>
Post Condiciones:	<ul style="list-style-type: none"> ➤ Mostrar información de la provincia y sus cantones ➤ Mostrar una descripción general del SIG ➤ Mostrar una ventana de ayuda del sistema

CUADRO 19: Caso de Uso # 9



✓ **Diagrama de Robustez**

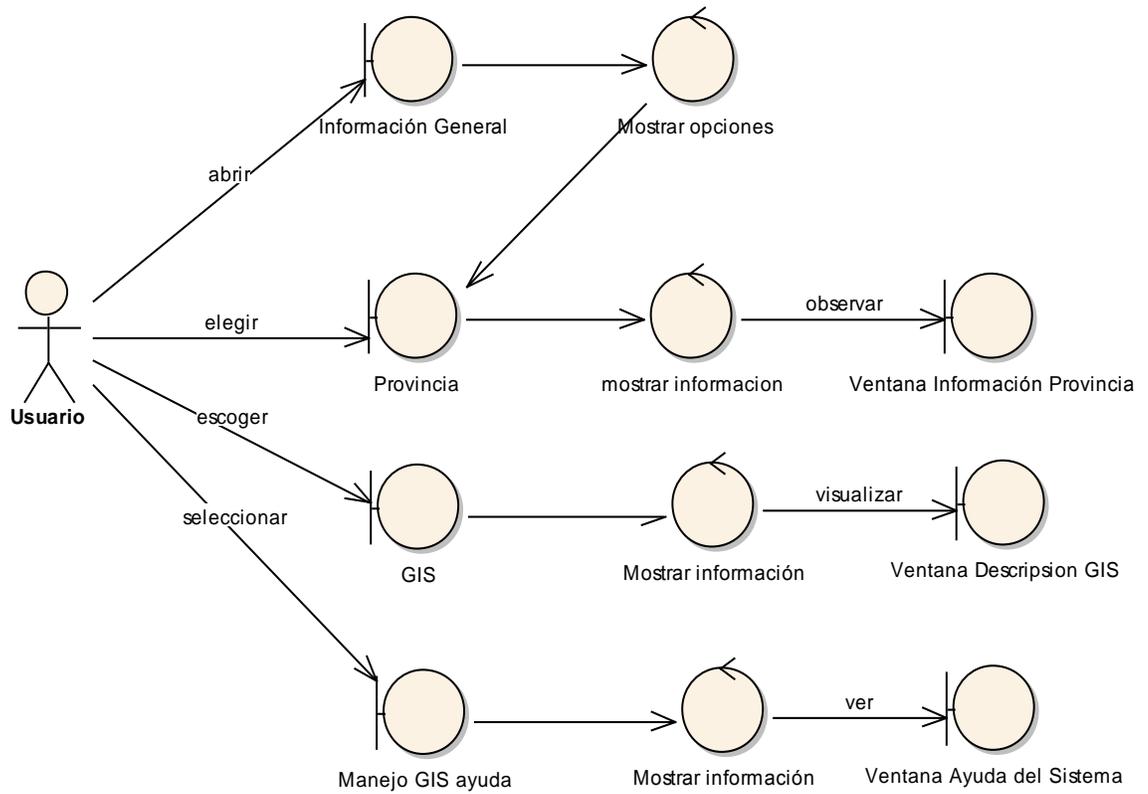


FIGURA 54: Diagrama de Robustez # 9



✓ Diagrama de Secuencia

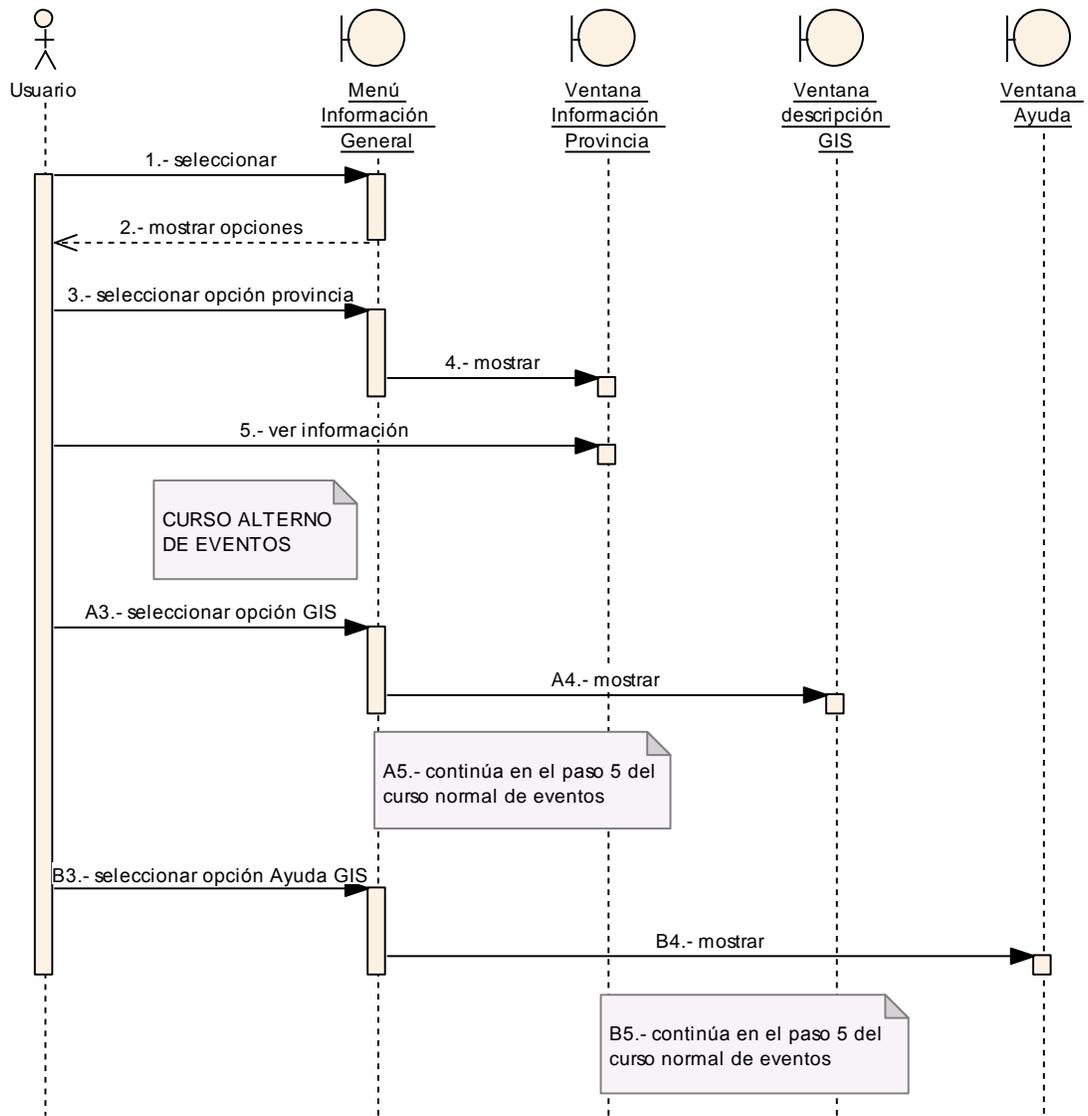


FIGURA 55: Diagrama de Secuencia # 9



7.4.3 Prototipo de Pantallas

Nombre de la Pantalla:	Ingreso al Sistema
Caso de Uso:	UC01
Referencia	P001



FIGURA 56: Plantilla instalación de la base de datos

Nombre de la Pantalla:	Ingreso al Sistema
Caso de Uso:	UC01, UC02, UC03, UC04, UC05, UC06, UC07, UC08
Referencia	P002

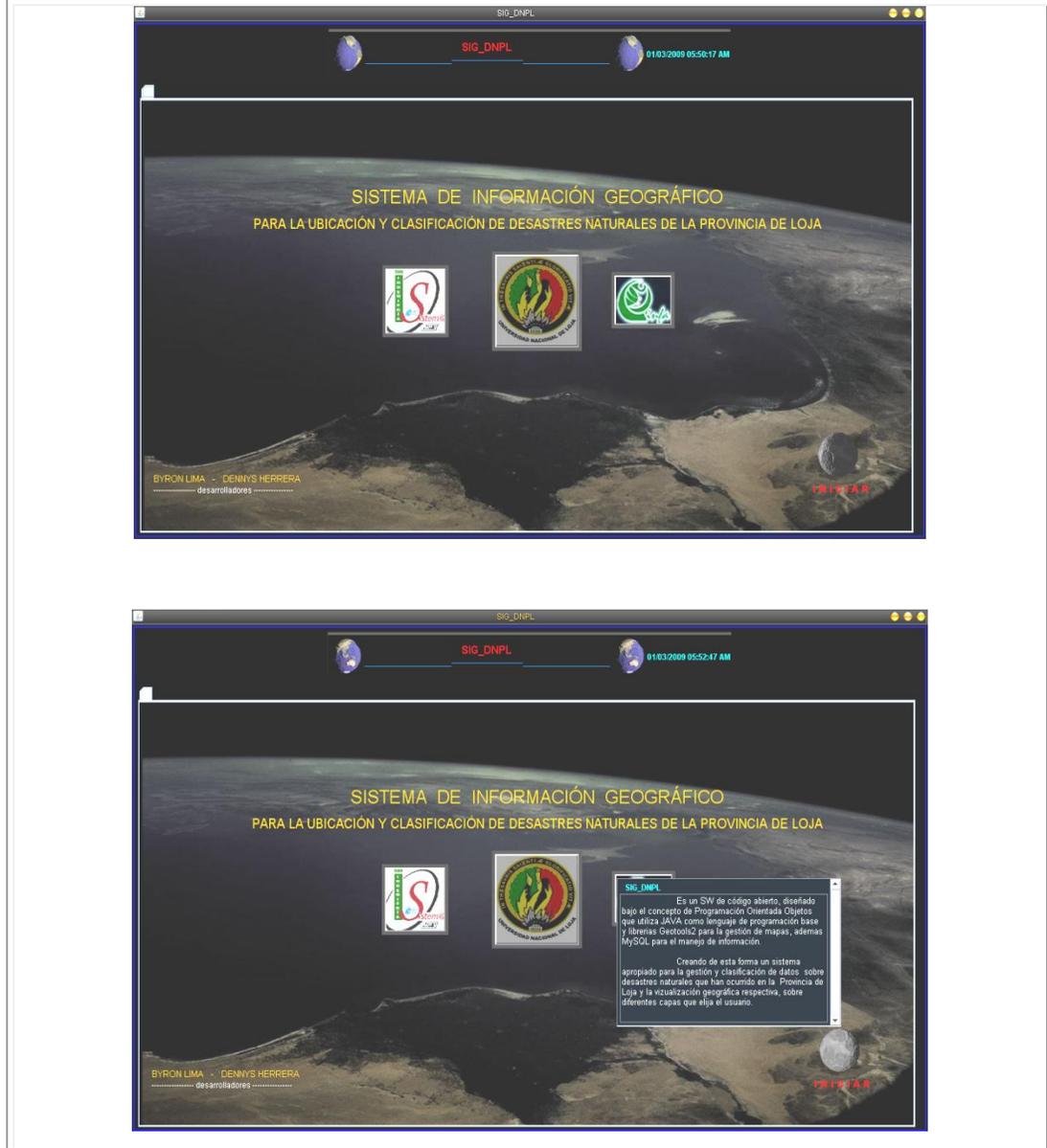


FIGURA 57: Plantilla ingreso al sistema



Nombre de Pantalla:	SIG para la ubicación y Clasificación de Desastres Naturales en la Provincia de Loja
Caso de Uso:	UC01, UC02, UC03, UC04, UC05, UC06, UC07, UC08
Referencia	P003



FIGURA 58: Plantilla Pantalla principal



Nombre de Pantalla:	Información General de la Provincia de Loja y Generalidades del Gis.
Caso de Uso:	UC08
Referencia	P004

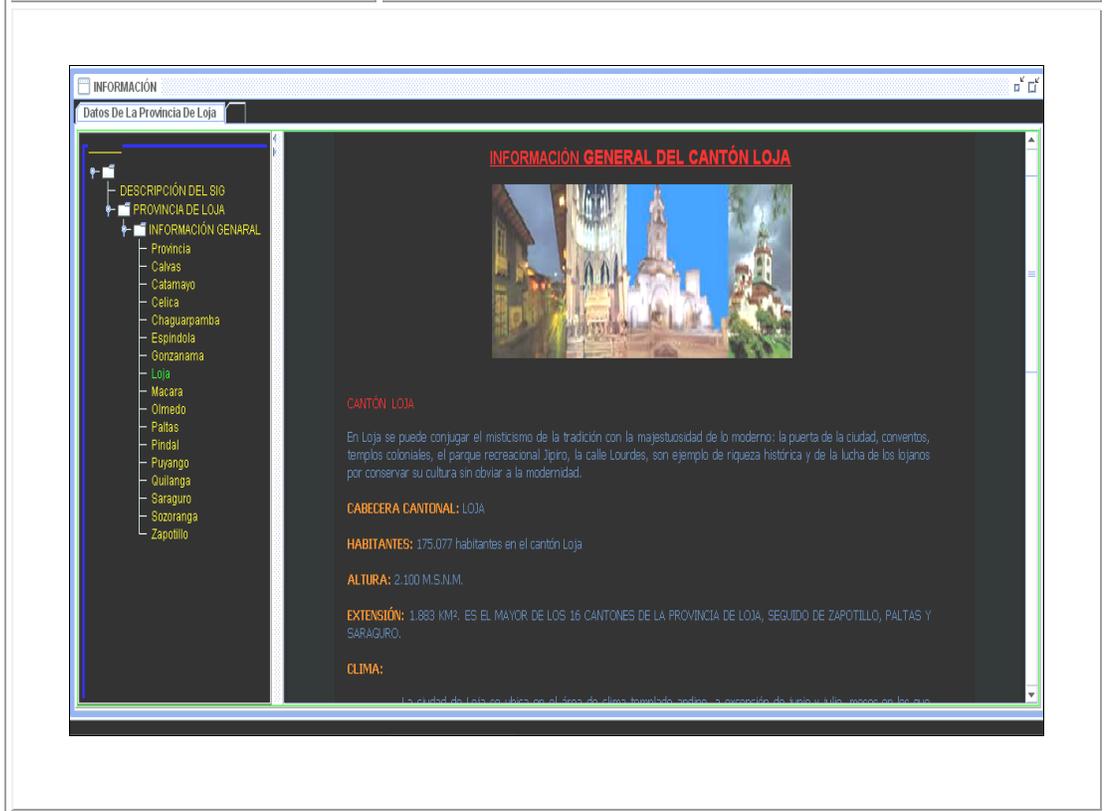


FIGURA 59: Plantilla Información General de la Provincia.



Nombre de la Pantalla:	Ayuda del SIG_DNPL
Caso de Uso:	UC08
Referencia	P005



FIGURA 60: Plantilla Ayuda SIG_DNPL

Nombre de la Pantalla:	Edición del Archivo Shape
Caso de Uso:	UC02
Referencia	P006



FIGURA 61: Plantilla Archivo Shape



Nombre de la Pantalla:	Visualizar Desastre Natural
Caso de Uso:	UC01
Referencia	P007

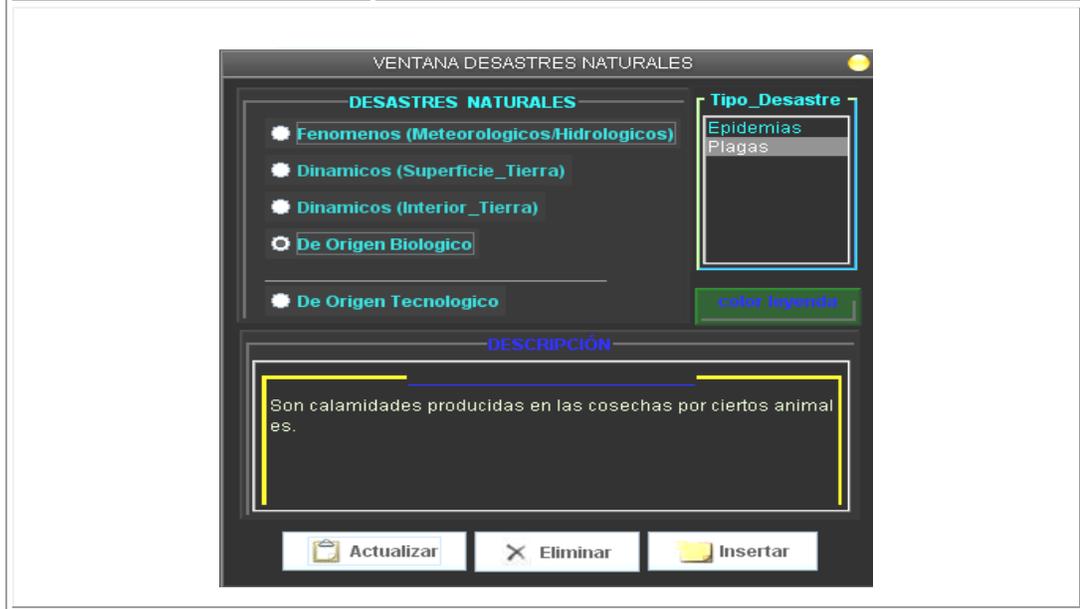


FIGURA 62: Plantilla Ventana Desastre Natural

Nombre de la Pantalla:	Modificar Desastre Natural
Caso de Uso:	UC01
Referencia	P008



FIGURA 63: Plantilla Modificar Desastre Natural

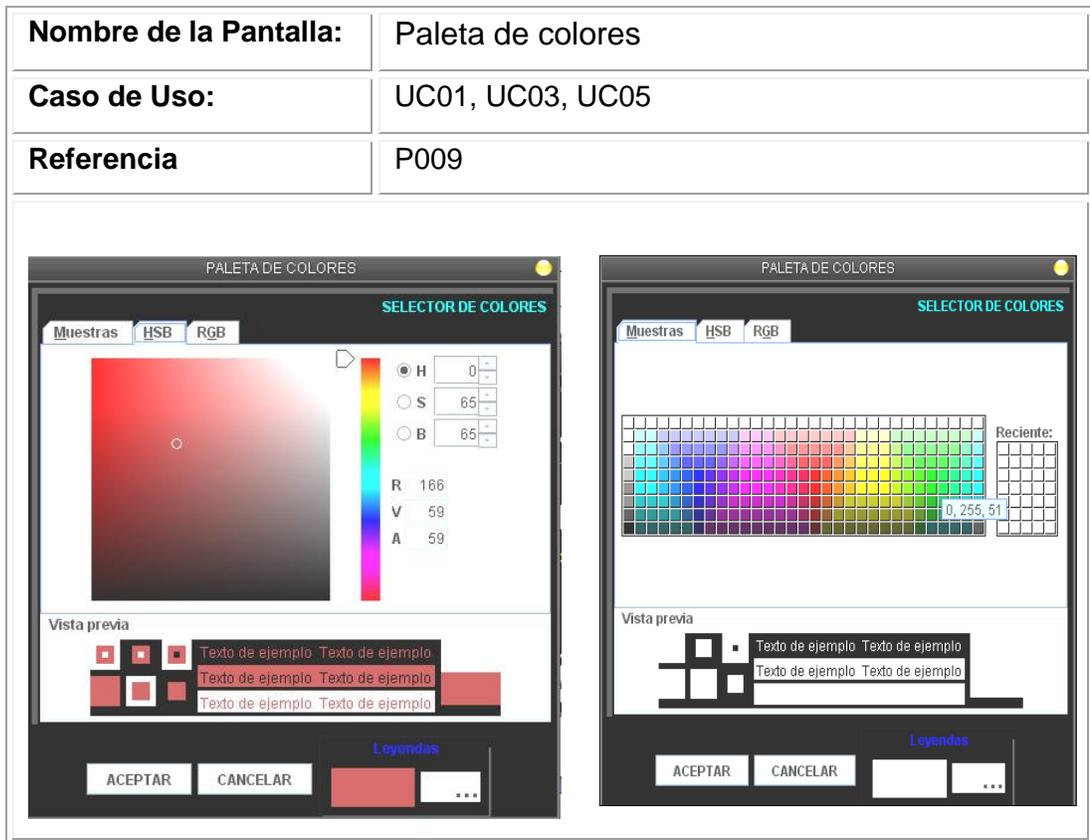


FIGURA 64: Plantilla Paleta de Colores



Nombre de la Pantalla:	Agregar nuevo desastre generado
Caso de Uso:	UC03
Referencia	P010

FIGURA 65: Plantilla Agregar Nuevo Desastre

Nombre de la Pantalla:	Convertir coordenadas geográficas a planas (UTM)
Caso de Uso:	UC03, UC05
Referencia	P011

FIGURA 66: Plantilla Coordenadas Geográficas



Nombre de la Pantalla:	Insertar informe de desastre
Caso de Uso:	UC03, UC05
Referencia	P012



FIGURA 67: Plantilla Insertar Informe Desastre

Nombre de la Pantalla:	Buscar Imágenes
Caso de Uso:	UC03, UC05
Referencia	P013

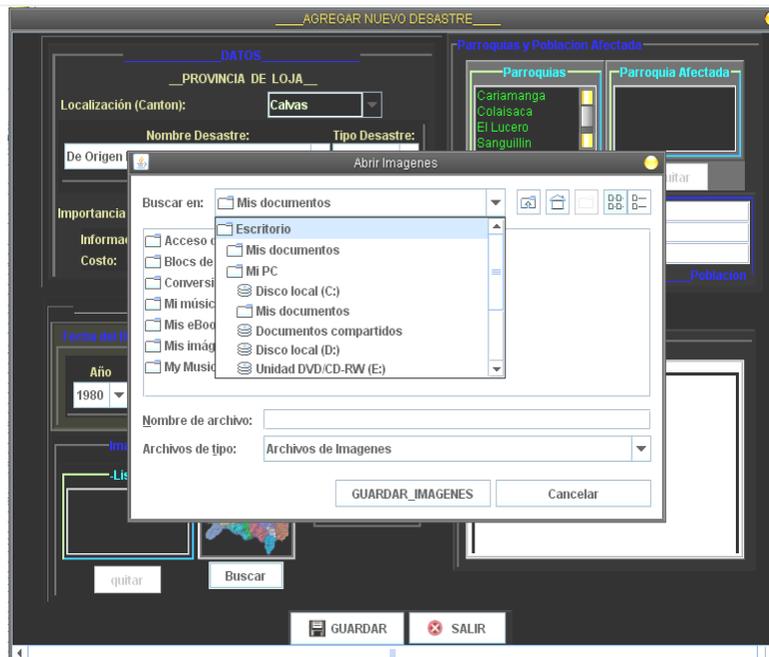


FIGURA 68: Plantilla Buscar Imagenes

Nombre de la Pantalla:	Buscar, Actualizar y Eliminar un Desastre Generado desde menú principal
Caso de Uso:	UC04, UC05
Referencia	P014

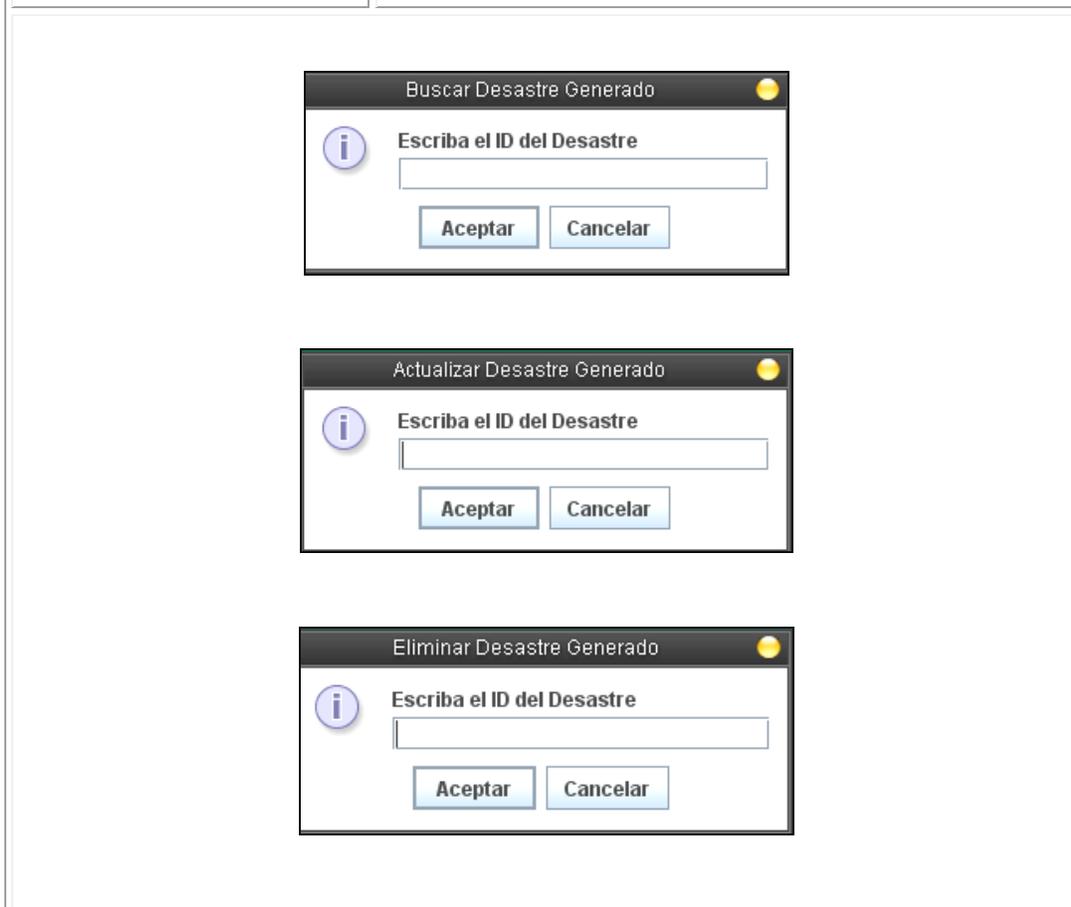


FIGURA 69: Plantilla Buscar, Actualizar y Eliminar Desastre



Nombre de la Pantalla:	Búsquedas
Caso de Uso:	UC06
Referencia	P015

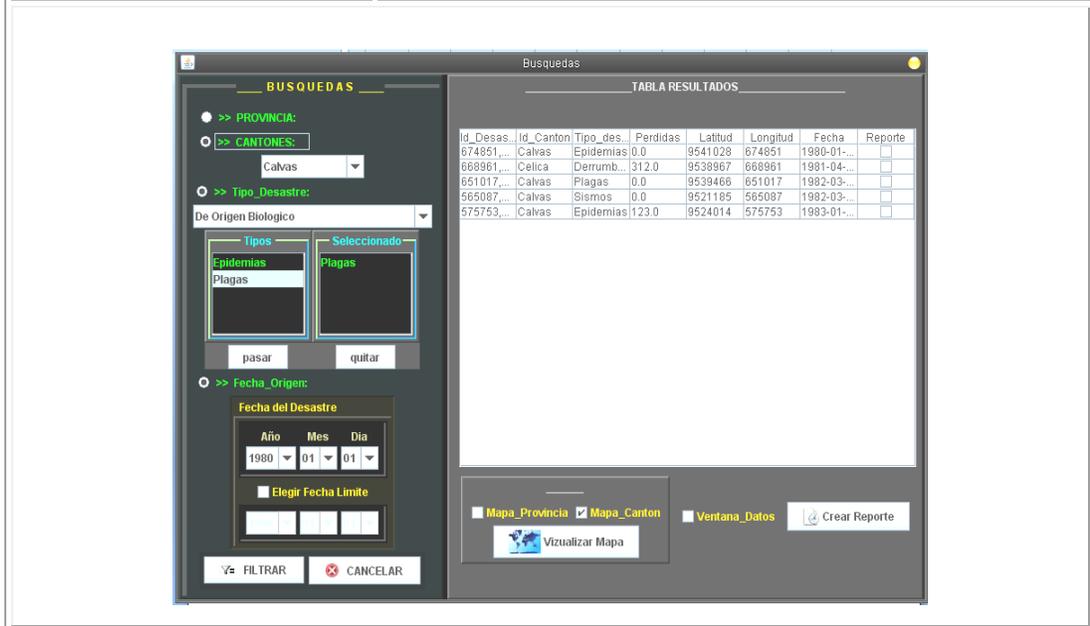


FIGURA 70: Plantilla Filtrar Información

Nombre de la Pantalla:	Reporte de Desastres filtrados
Caso de Uso:	UC07
Referencia	P016



FIGURA 71: Plantilla Reporte Desastre

Nombre de la Pantalla:	Visualizar puntos filtrados en el mapa
Caso de Uso:	UC06
Referencia	P017

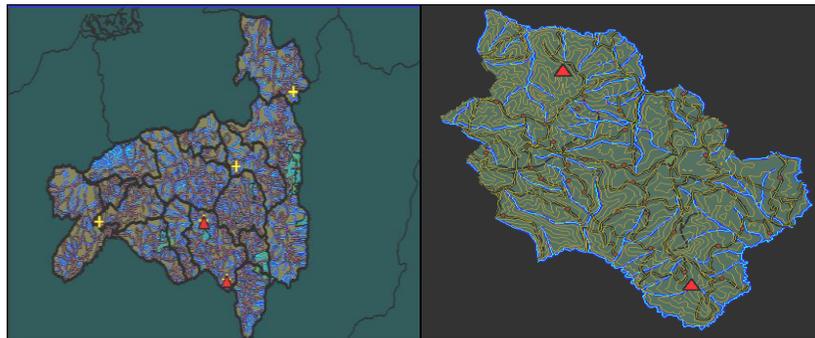
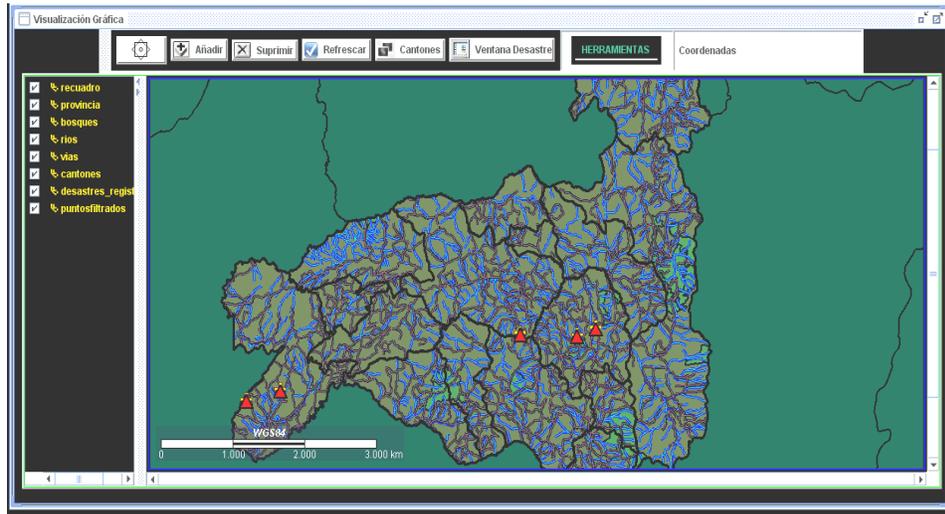


FIGURA 72: Plantilla Visualizar puntos Filtrados

Nombre de la Pantalla:	Visualizar Información de Desastre Generado
Caso de Uso:	UC04
Referencia	P018

DESASTRE NATURAL GENERADO EN LA PROVINCIA DE LOJA

EFFECTOS

- Obstrucción en la red vial principal que va desde Loja a Catamayo durante algunos días
- Se producen daños en el sector Agropecuario en las fincas en donde se produce el deslizamiento,

DATOS

PROVINCIA:	LOJA
Localización (Cantón):	Loja
Información Recogida Por:	Juan Pelaez
Fecha de Origen:	2008-04-04
Importancia del Evento:	Alta
Costo de Perdidas:	\$ 20000.0

UTM Este X: 698110 m

UTM Norte Y: 9548810 m

Coordenadas

Nombre Desastre: Dinamicos (Superficie_Tierra)

Tipo Desastre: Derrumbes

DESCRIPCIÓN DESASTRE

Imágenes del Desastre

DSC00005.JPG



←
→
🖨
🖼
Galeria

Parroquias y Población Afectada

Parroquias

Loja

El Cisne

Número Damnificados: 10

Número de Muertos: 3

Número de Heridos: 2

Población

REPORTE

Actualizar

Borrar

Información General

SALIR

FIGURA 73: Plantilla Desastre Natural Generado

Nombre de la Pantalla:	Zoom Imagen
Caso de Uso:	UC03, UC04, UC05
Referencia	P019

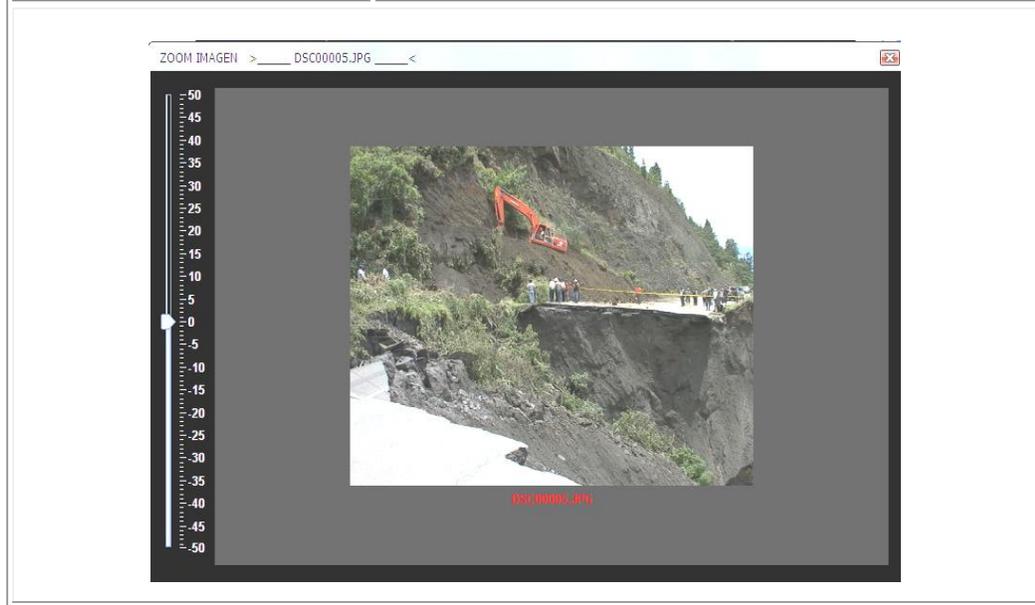


FIGURA 74: Plantilla Zoom Imagen

Nombre de la Pantalla:	Visualizar Informe de Desastre Generado
Caso de Uso:	UC04
Referencia	P020

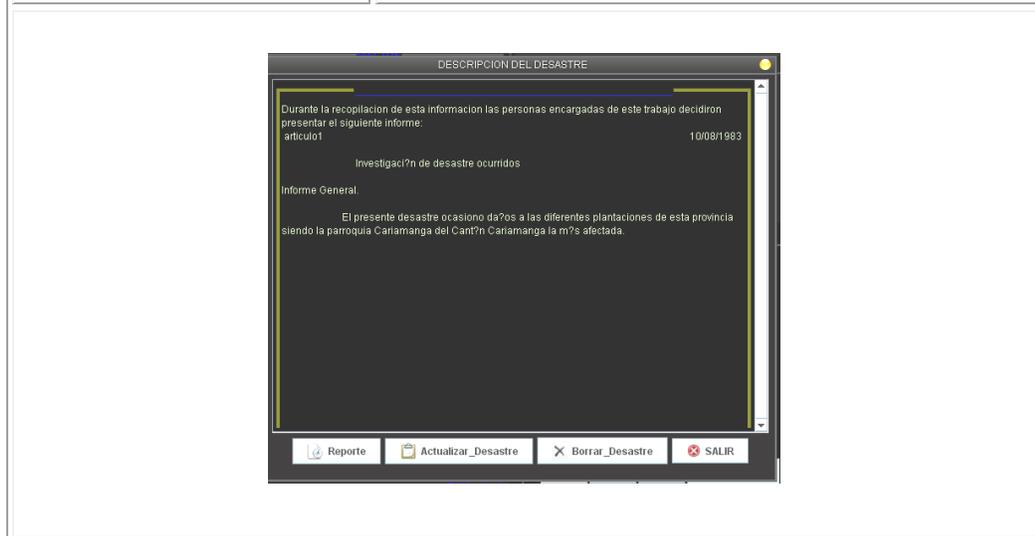


FIGURA 75: Plantilla Visualizar Informe Desastre

Nombre de la Pantalla:	Galería de Imágenes
Caso de Uso:	UC04
Referencia	P021



FIGURA 76: Plantilla Galería de Imágenes



Nombre de la Pantalla:	Reporte Desastre Generado
Caso de Uso:	UC07
Referencia	P022

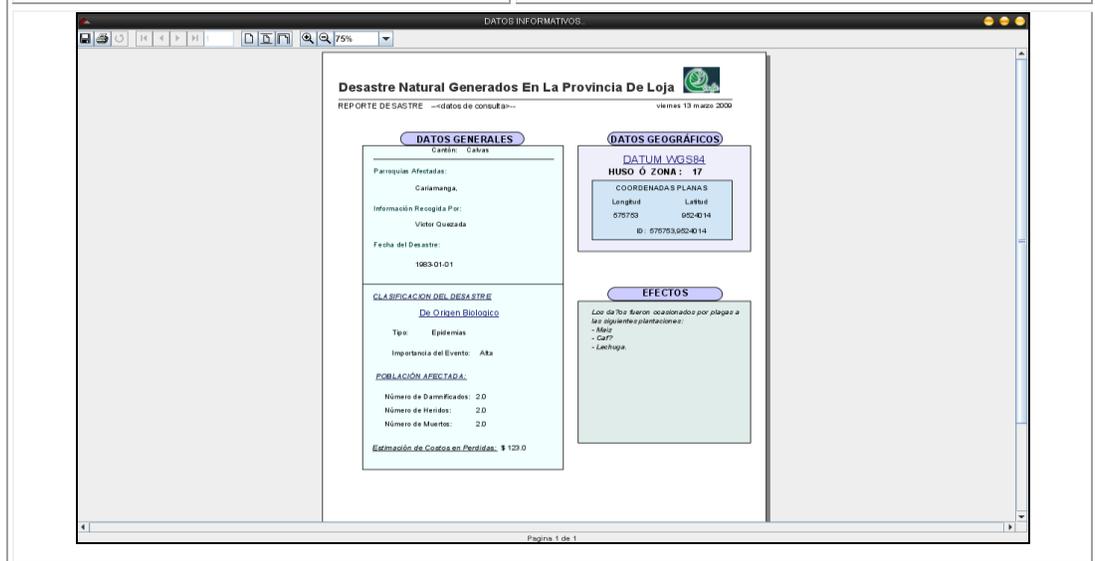


FIGURA 77: Plantilla Reporte Desastre Generado

Nombre de la Pantalla:	Reporte Informe General del Desastre
Caso de Uso:	UC07
Referencia	P023

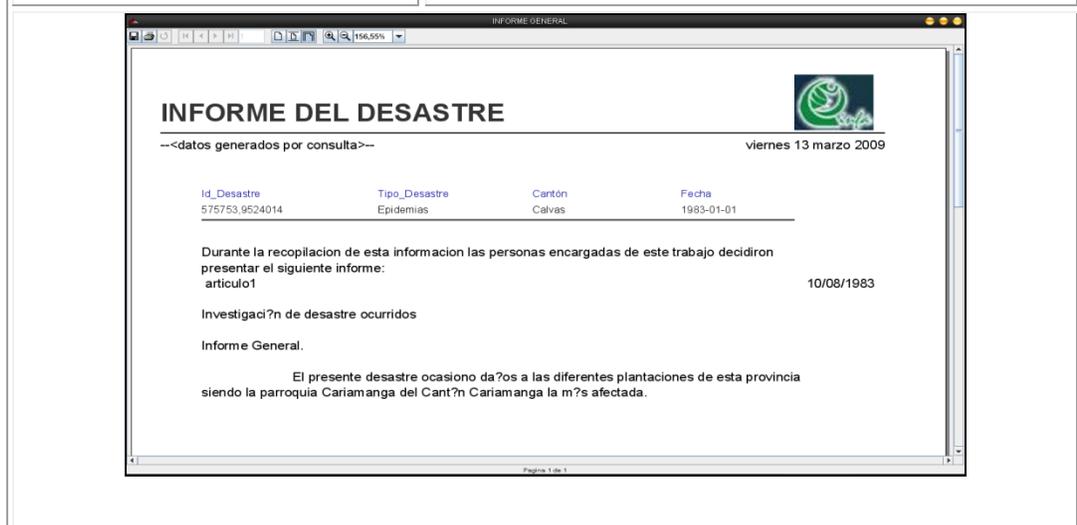


FIGURA 78: Plantilla Reporte Informe General del Desastre

Nombre de la Pantalla:	Reporte Imagen del desastre
Caso de Uso:	UC05
Referencia	P024

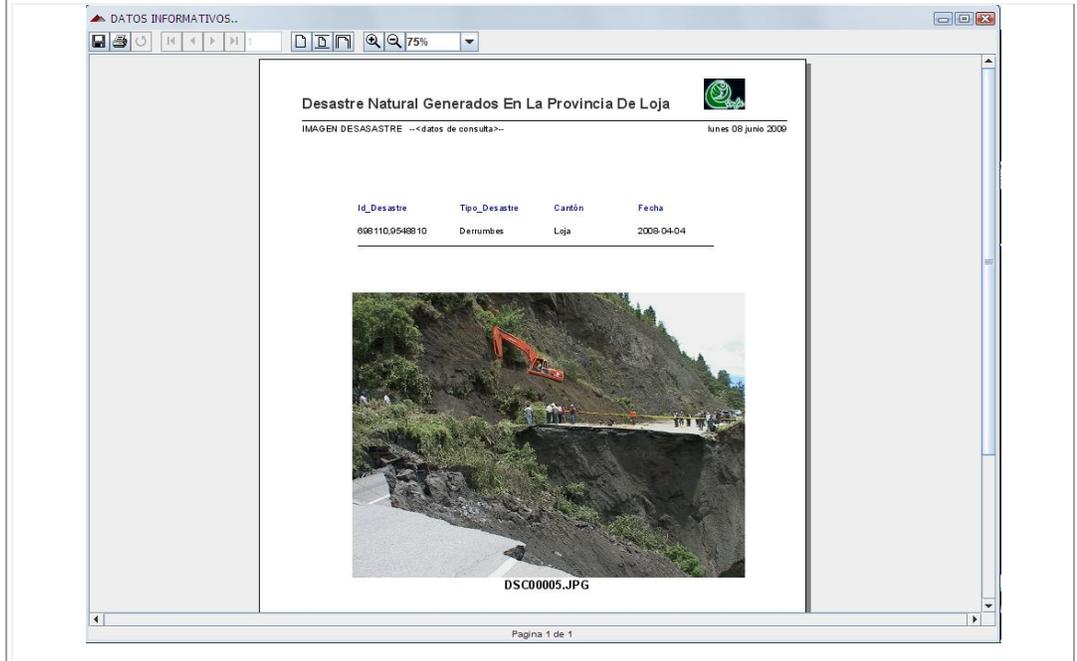


FIGURA 79: Plantilla Reporte Imagen

Nombre de la Pantalla:	Leyenda Capas
Caso de Uso:	UC02
Referencia	P025



FIGURA 80: Plantilla Leyenda Capas

Nombre de la Pantalla:	Resaltar Cantón
Caso de Uso:	UC03
Referencia	P026

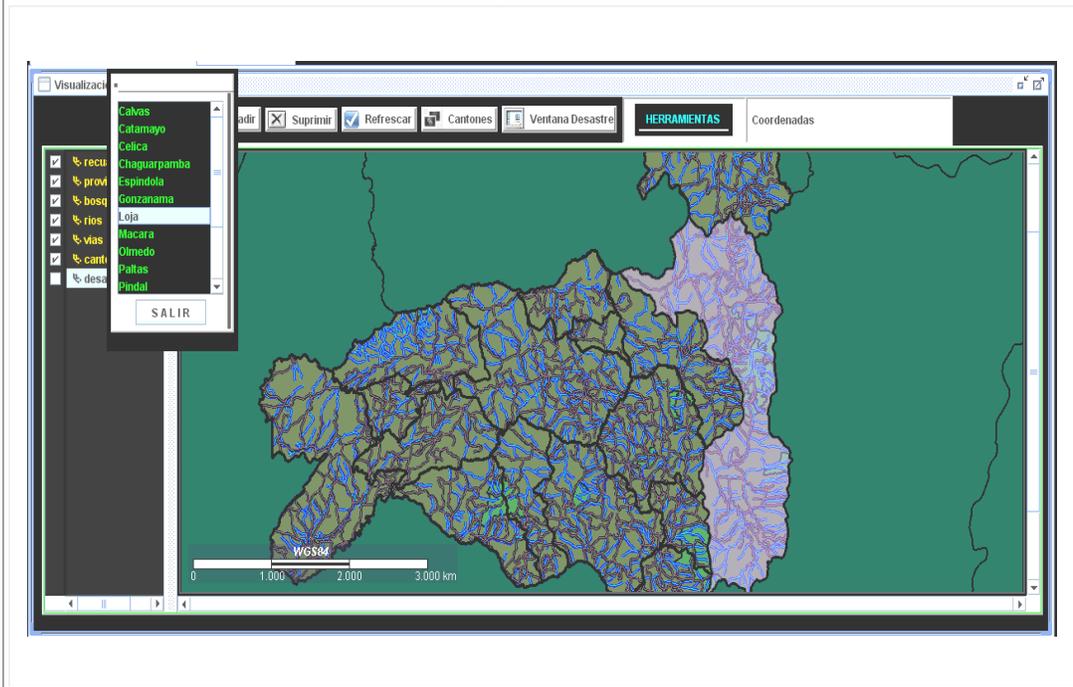


FIGURA 81: Plantilla Resaltar Cantón

Nombre de la Pantalla:	Resaltar Desastre Generado
Caso de Uso:	UC06
Referencia	P027

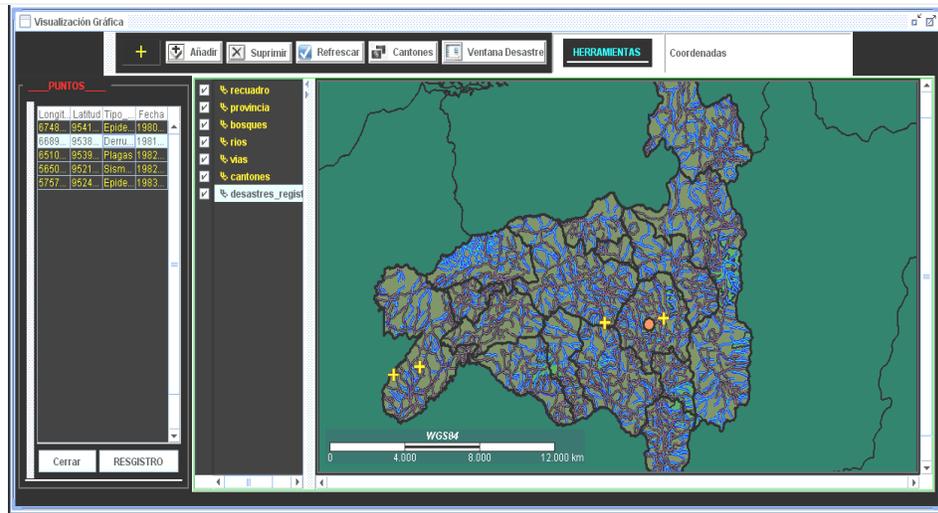


FIGURA 82: Plantilla Resaltar Desastre Generado

Nombre de la Pantalla:	Herramientas para manipular capas
Caso de Uso:	UC02
Referencia	P028

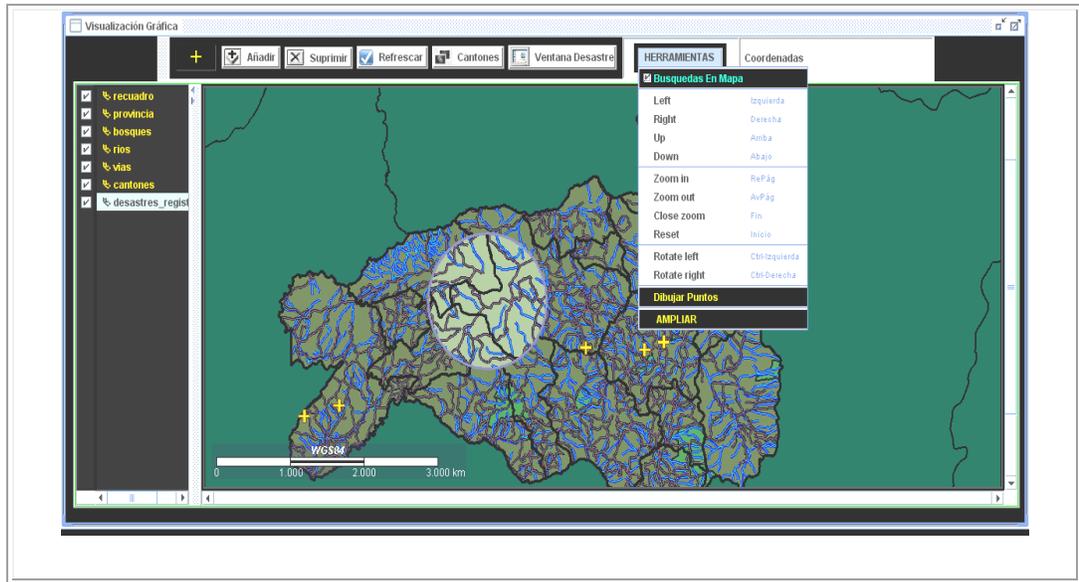


FIGURA 83: Plantilla Herramientas de Manipulación de Capas

Nombre de la Pantalla:	Datos Filtrados de la ultima búsqueda
Caso de Uso:	UC06
Referencia	P029

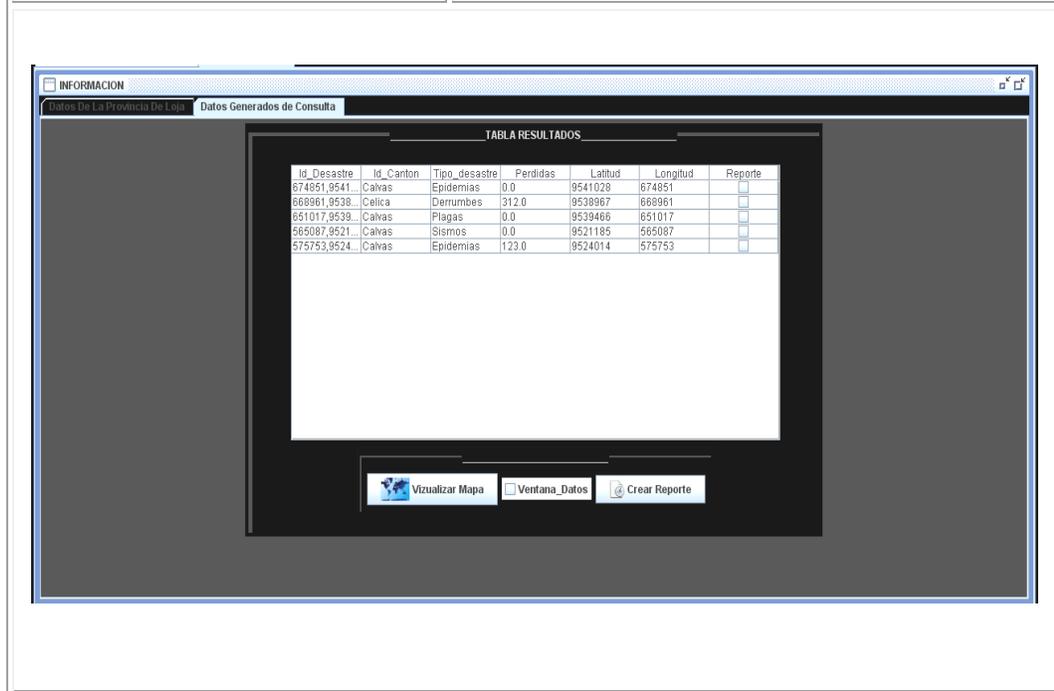


FIGURA 84: Plantilla Datos Filtrados de la última Búsqueda



7.4.4 Diagrama de Paquetes

El diagrama de paquete se encuentra comprendido por los siguientes paquetes:

- Paquete SIG_DNPL
- Paquete SIG_DNPL.modelo
- Paquete SIG_DNPL.negocio
- Paquete SIG_DNPL.vistas

Cada paquete se describe a continuación, con sus subpaquetes y clases que la conforman:

Paquete SIG_DNPL

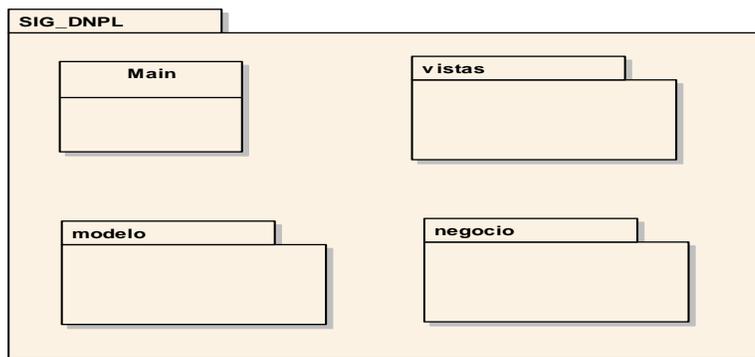


FIGURA 85: Paquete SIG_DNPL



Paquete SIG_DNPL.negocio

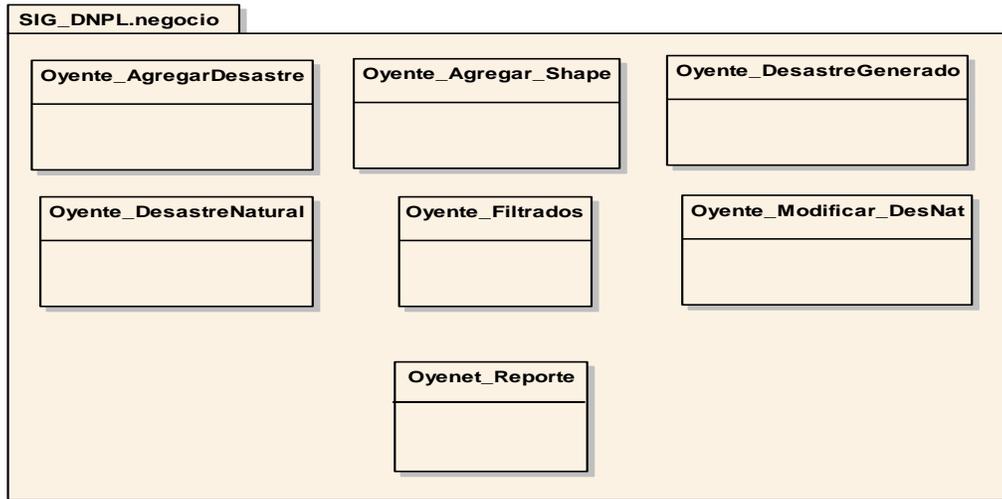


FIGURA 86: Paquete SIG_DNPL.negocio
Paquete SIG_DNPL. Modelo

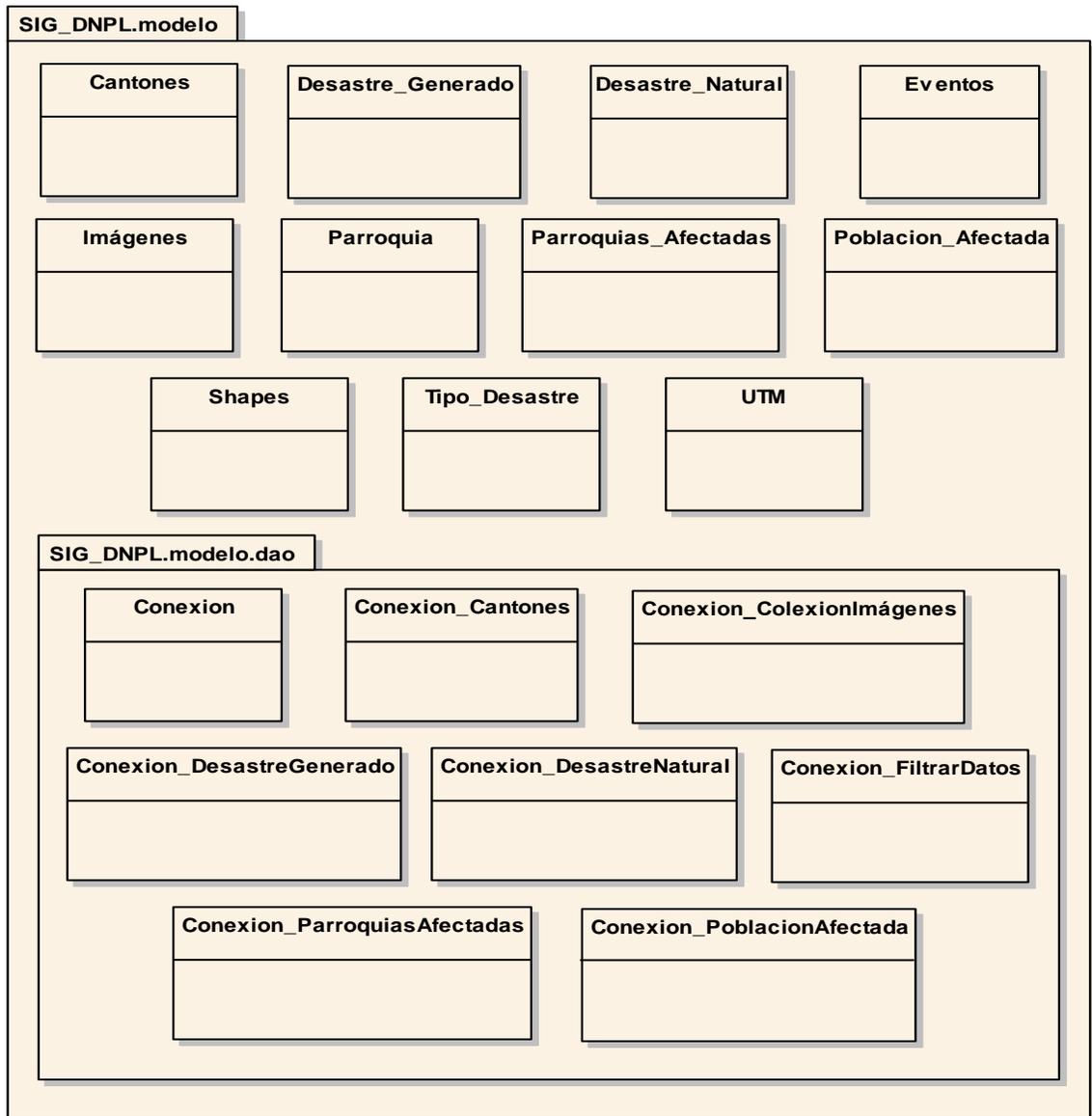


FIGURA 87: Paquete SIG_DNPL.modelo

Paquete SIG_DNPL.vistas

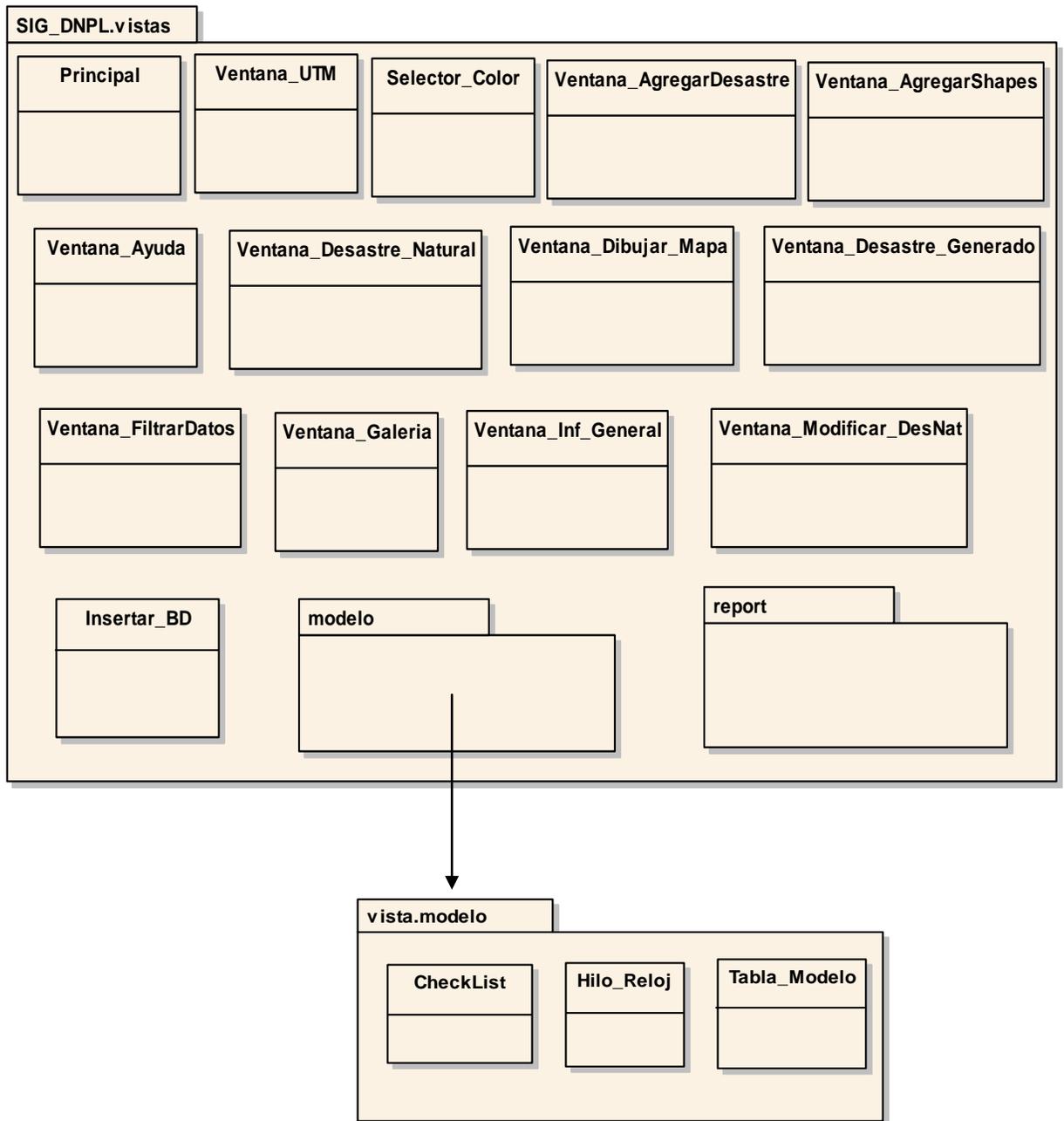


FIGURA 88: Paquete SIG_DNPL.vistas



7.4.5 Diagrama de Clases



7.4.6 Diagrama Entidad-Relación

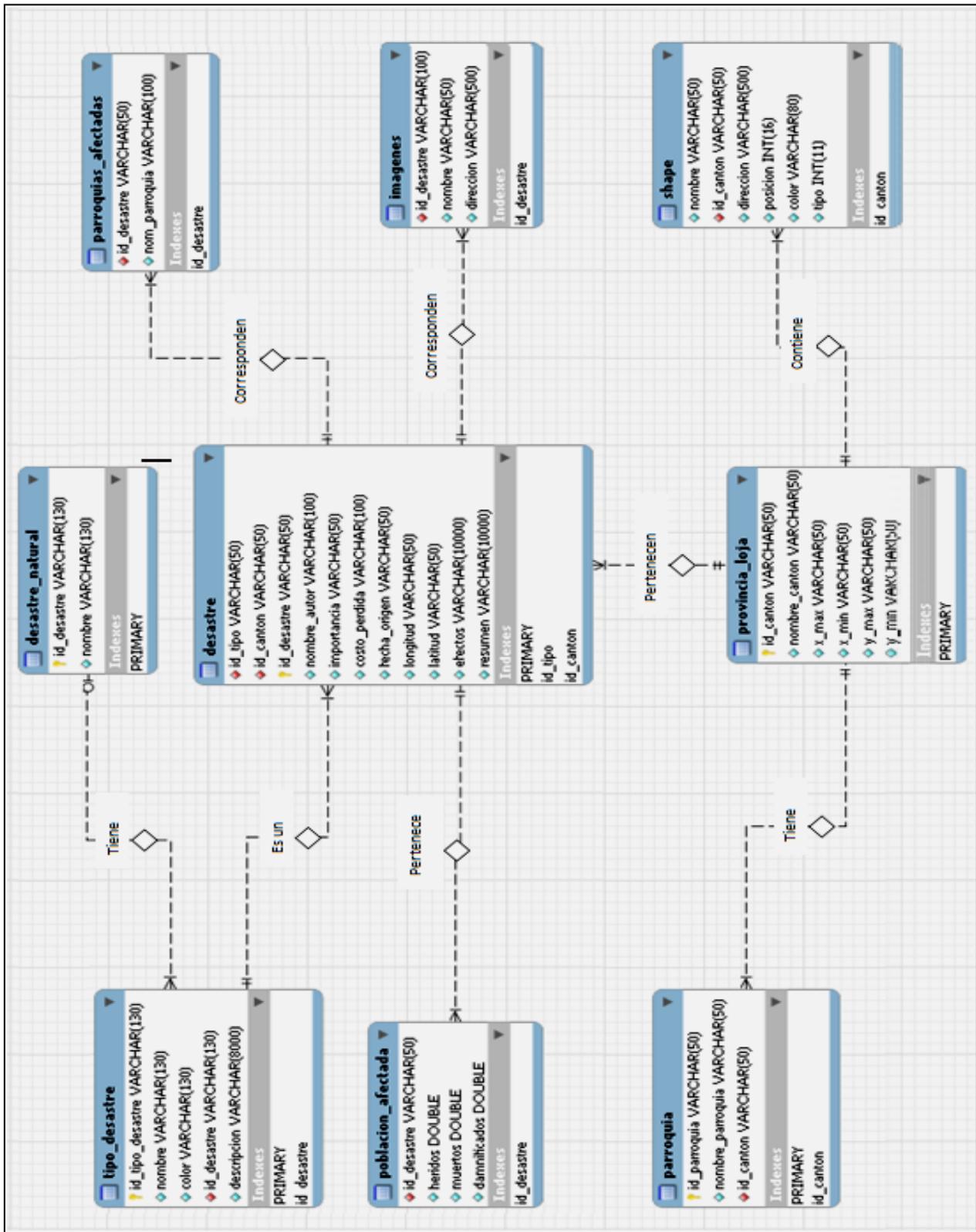


FIGURA 87: Diagramas Entidad Relación



7.5 Plan de Validación

El plan de validación es la etapa en la que se evalúa la funcionalidad y eficiencia del sistema frente al usuario final que va a estar en contacto directo con la aplicación, y que es quien valida los procesos que la aplicación realiza.

7.5.1 Tipos de Pruebas Aplicadas al Sistema

Los tipos de pruebas aplicadas al sistema “SIG (Sistema de Información Geográfico) para la Ubicación y Clasificación de Desastres Naturales de la Provincia de Loja” son:

- Pruebas de Funcionalidad
- Pruebas de Aceptación
- Pruebas de Usabilidad

7.5.1.1 Pruebas de Funcionamiento

Este tipo de pruebas fueron aplicadas a la funcionalidad del sistema dentro de los procesos que este maneja (*Ver Cuadro de Resultado de pruebas, pág. 197*), que son los siguientes:

- ✓ Administración en la clasificación de Desastre Naturales.
- ✓ Manipulación de las capas: Provincia, Cantones, Parroquias, ríos, vías, cabeceras parroquiales, desastres generados, etc.
- ✓ Utilización de las herramientas de navegación (zoom, giros, desplazar, ampliar) sobre las capas.
- ✓ Cambio del color a una capa seleccionada.
- ✓ Búsqueda físicas dentro del mapa, de un desastre generado haciendo clic en un punto marcado como desastre en el mapa
- ✓ Búsqueda precisa de desastres generados de acuerdo a una clasificación.
- ✓ Búsquedas directas desde una tabla de consulta.
- ✓ Manejo y control de opciones (provincia, cantones, tipo de desastre, fecha) para realizar búsquedas específicas.
- ✓ Control de errores por incompatibilidad en los datos
- ✓ Control y creación de reportes en base a la consulta realizada.



- ✓ Introducción de un nuevo desastre al sistema
- ✓ Control en el ingreso de desastres generados de acuerdo al área que tiene cada cantón
- ✓ Control en el ingreso del tipo de coordenadas al sistema
- ✓ Conversión de coordenadas de tipo geográficas a coordenadas planas (UTM)
- ✓ Introducción de imágenes desde cualquier directorio.
- ✓ Tiempos de carga de la información sobre cada desastre.
- ✓ Tiempos de respuesta de acceso a la BD.
- ✓ Facilidad en la interpretación para el manejo de los componentes gráficos
- ✓ Facilidad de navegación en el sistema.

7.5.1.2 Pruebas de Aceptación

Este tipo de pruebas son aplicadas para verificar la funcionalidad total de la aplicación, la misma que se las realizo a los usuarios del sistema empleando una encuesta (*Ver Anexo 2: Encuestas Realizadas*) y a través de la manipulación del sistema, para a su vez probar el correcto funcionamiento de todos los procesos.

7.5.1.3 Pruebas de Usabilidad

Pruebas aplicadas para verificar la utilidad del sistema y la robustez del mismo (*Ver Anexo 2: Encuestas Realizadas*):

- ✓ Interfaz amigable
- ✓ Facilidad de navegación
- ✓ Facilidad de manejo de herramienta y controles en los mapas
- ✓ Facilidad de realización de consultas y reportes.
- ✓ Seguimiento de aplicaciones
- ✓ Interpretación de errores

7.5.2 Fase de Validación

La fase de validación de acuerdo con la planificación de nuestro proyecto se realizo el día 24 de Marzo del 2009 y de acuerdo con los objetivos se la llevó a cabo



en el CINFA de la Universidad Nacional de Loja, teniendo como responsables a los directivos y personal técnico de la institución.

Dentro de este proceso de validación las pruebas fueron aplicadas a las siguientes personas como usuarios del “SIG para la ubicación y clasificación de Desastres Naturales en la Provincia de Loja”:

- Ing. Aníbal Gonzales (Director del CINFA)
- Ing. José Merino (Técnico del CINFA)
- Ing. Fabián Sotomayor (Técnico del CINFA)

De lo que nos da un total de tres “encuestas realizadas”³⁶. La aplicación de las encuestas se realizó luego que los usuarios utilizaran el sistema y manipularan todos los procesos que este realiza, donde entre otras cosas se verificó velocidad de respuesta, facilidad de manejo, controles del sistema, etc.

Para representar los valores de las entrevistas a continuación mostramos los rangos de evaluación:

E: Excelente.

M: Muy Buena.

B: Buena.

R: Regular

T: Total

El rol va a estar ejecutado por los usuarios que intervienen en el sistema, y la valoración de acuerdo a un rango definido por las encuestas realizadas es de 0 a 3.

A continuación se presenta una tabla de valores con los resultados de las pruebas realizadas en esta fase.

Para mejor entendimiento de la representación del cuadro (Cuadro 19), veremos que los valores van a estar dados sobre el número de encuestas (en este caso **3**). Por ejemplo se va a explicar con el primer parámetro evaluado (funcionalidad): **Acceso a menús principales** tiene como resultado **0** encuestas para el valor de **E (Excelente)**, **3** encuestas para el valor de **M (Muy Buena)**, **0** encuestas para el valor de **B (Buena)**, **0** encuestas para el valor de **R (Regular)**; sumando estos

³⁶ Ver Anexo 2: Encuestas Realizadas



valores se registra en **T (Total)** el valor de **3**; este mismo mecanismo se sigue para las siguientes parámetros.

FUNCIONALIDAD	E	M	B	R	T
Acceso a menús principales	0	3	0	0	3
Acceso a las capas mediante los menús principales	1	1	1	0	3
Visualización de las capas geográficas que conforman la Provincia de Loja.	1	1	1	0	3
Visualización de coordenadas planas (UTM) en el mapa.	1	1	1	0	3
Aplicación de las herramientas de consulta (acercar, alejar, giros, desplazar, ampliar) sobre las capas.	1	1	1	0	3
Manipulación de las capas (insertar, eliminar, actualizar)	1	1	1	0	3
Asignación de colores específicos para cada capa.	1	1	1	0	3
Acceso a la información de desastres generados mediante las opciones que ofrecen los menús principales.	1	2	0	0	3
Interacción directa con el mapa para realizar consultas sobre desastres generados en la Provincia de Loja	1	2	0	0	3
Consultas específicas (por cantón, por tipo de desastre, por fecha de origen) de los desastres generados registrados en la base de datos.	1	2	0	0	3
Visualización de la tabla que almacena la última consulta realizada al sistema.	1	2	0	0	3
Generación de reportes en base a la consulta realizada.	1	2	0	0	3
Visualización de los mapas (cantón o provincia) según la consulta seleccionada sobre los desastres generados.	1	2	0	0	3
Visualización de imágenes e información relacionada para cada desastre	1	2	0	0	3
Actualización de la información almacenada sobre desastres.	0	3	0	0	3
Ingreso de un nuevo desastre generado al sistema	1	1	1	0	3
Conversión de coordenadas geográficas a coordenadas planas (UTM)	1	1	1	0	3

Descripción paso a paso del proceso utilizado para la conversión	1	1	1	0	3
Introducción de imágenes para dicho desastre.	1	1	1	0	3
Incorporación de un informe general para el desastre generado	1	1	1	0	3
Generación de reportes en base a la información obtenida para cada desastre generado.	1	2	0	0	3
Generación de reportes en base a un informe o una imagen seleccionada	1	2	0	0	3
Impresión del reporte generado	1	2	0	0	3
Visualización de los diferentes tipos de desastres naturales	1	2	0	0	3
Manipulación (insertar, actualizar, eliminar) de la información sobre desastres naturales que pueden ocurrir	0	3	0	0	3
Manejo de la ayuda que ofrece el sistema para su correcta utilización	1	1	1	0	3
Descripción general de la Provincia y sus cantones	1	1	1	0	3
Visualización de mensajes informativos	2	1	0	0	3
Resultado	26	45	13	0	84

CUADRO 20. Cuadro de resultados de pruebas

7.5.3 Análisis del Resultado de Pruebas

Del cuadro de resultado de pruebas (Cuadro 19) se puede concluir que existe 31% de una aceptación **Excelente**, 54% de aceptación **Muy Buena**, y un 15% de aceptación **Buena** en las funciones del sistema.

Así al evaluar toda la tabla con las pruebas realizadas, se obtienen los siguientes resultados:

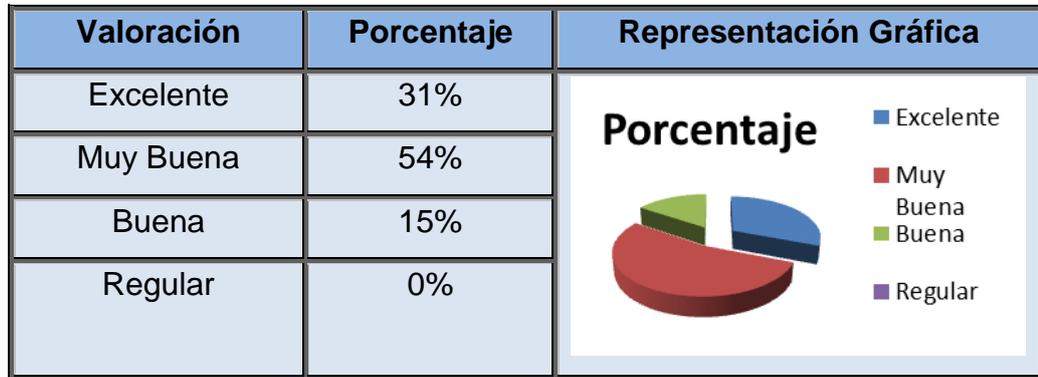
E (Excelente) = 26

M (Muy Buena) = 45

B (Buena) = 13

R (Regular) = 0

Para un mejor entendimiento, a continuación se presenta un porcentaje correspondiente a la valoración de las funcionalidades del sistema:



CUADRO 21. Cuadro de Porcentajes

Como se observa en el Cuadro de Porcentajes (Cuadro 20) el 54% de la aceptación **Muy Buena** es el de mayor porcentaje, esto debido a que dentro de las pruebas los encuestados no encontraron errores en la ejecución del sistema sino más bien se logro obtener nuevos requerimientos para futuras investigaciones. Además los usuarios realizaron observaciones en cuanto a la presentación de mensajes y representación de coordenadas en el mapa, lo que se tomó en cuenta para corregir y mejorar el sistema.

Finalizando las pruebas se concluyó que el “SIG para la ubicación y clasificación de Desastres Naturales en la Provincia de Loja” si cumple con todos los requerimientos planteados en el proyecto.



8. VALORACIÓN TÉCNICO-ECONÓMICA

VALORACIÓN TECNICO – ECONOMICA

El sistema de Información Geográfica para la ubicación y Clasificación de Desastres Naturales de la Provincia de Loja contribuye al desarrollo de la ciudad y provincia en el campo de los desastres naturales, es decir en la visualización, ingreso y manipulación de desastres naturales generados en la Provincia de Loja.

El software final cumple con todas las necesidades y requerimientos expuestos al inicio del proyecto, debido a que se utilizaron métodos y herramientas para su desarrollo de muy buena calidad.



Los costos para el desarrollo del sistema son muy bajos, debido a que se utilizó totalmente software libre como lo es el lenguaje de programación JAVA y Base de datos Mysql, así como las librerías geotools2 y JasperReport, todas con licencia GNU/GLP.

Finalmente los costos de fabricación en su totalidad por los dos desarrolladores son:

RECURSOS HUMANOS				
DESCRIPCION	CANTIDAD	NUMERO HORAS	VALOR UNITARIO	VALOR TOTAL
Desarrolladores: <ul style="list-style-type: none"> • Dennys Herrera. • Byron Lima. 	-----	-----	-----	-----
Expertos: <ul style="list-style-type: none"> • Docentes • Personal Técnico del CINFA • Director del CINFA 	-----	-----	-----	-----
Asesor profesional	1	40	\$ 10	\$ 400
Subtotal				\$ 400

CUADRO 22. Recursos Humanos

RECURSOS TECNICOS.				
DESCRIPCION	CANTIDAD	NUMERO HORAS	VALOR/ UNITARIO	VALOR/ TOTAL
Software:				
Lenguaje de programación: - JAVA SE 6	1	-----	-----	Gratuito
Gestor de Base de Datos: - MySQL 5.0	1	-----	-----	Gratuito
NETBEANS IDE 6.1	1	-----	-----	Gratuito
UML para desarrollo de	1	-----	-----	Gratuito



diagramas de clases del sistema				
<u>Comunicación:</u>				
Internet	-----	500	\$ 0.80	\$ 400
Libros y de mas archivos de consulta.				
			Subtotal	\$ 400

CUADRO 23. Recursos Técnicos

RECURSOS MATERIALES.				
DESCRIPCION	CANTIDAD	NUMERO HORAS	VALOR/UNITARIO	VALOR/TOTAL



<u>Materiales de Oficina.</u>				
▪ Cartuchos para la impresora.	15		\$ 6.00	\$ 90.00
▪ Flash Memory.	1		\$ 15.00	\$ 15.00
▪ Perforadora.	1		\$ 4.00	\$ 4.00
▪ Grapadora	1		\$ 3.00	\$ 3.00
▪ Caja de Grapas	1		\$ 1.00	\$ 1.00
▪ Caja de Clips	1		\$ 1.00	\$ 1.00
▪ Caja de Cds.	1		\$ 15.00	\$ 15.00
▪ Esferográficos	6		\$ 0.25	\$ 1.50
▪ Anillados.	5		\$ 1.50	\$ 7.50
▪ Carpetas.	6		\$ 0.25	\$ 1.00
▪ Copias.	500		\$ 0.02	\$ 10.00
▪ Resma de papel bond	10		\$ 3.50	\$ 35.00
<u>Servicios Básicos.</u>				
▪ Transporte.				\$ 100.00
▪ Electricidad.				\$ 80.00
▪ Agua.				\$ 10.00
▪ Teléfono.				\$ 30.00
			Subtotal	\$ 404.00
			Imprevistos	\$ 100.00
			TOTAL	\$1304.00

CUADRO 24. Recursos Materiales



9. CONCLUSIONES

CONCLUSIONES:

Al culminar el desarrollo de nuestro proyecto de tesis, hemos llegado a las siguientes conclusiones:

- La información referente a desastres naturales generados en la provincia de Loja que se encontraba aislada en distintos tipos de archivos (hojas de Word y Excel), se administra y almacena de manera más eficiente al ser ingresada en



el SIG para la Ubicación y Clasificación de Desastres Naturales en la Provincia de Loja.

- La utilización íntegra de tecnología JAVA como lenguaje de programación y MySQL como gestor de base de Datos (ambas de distribución gratuita) frente a C# de la plataforma .NET (con licencias de software pagadas) nos permitió obtener un importante ahorro en cuanto a costes de licencias de software ya que su distribución es gratuita.
- El uso del IDE NetBeans con el plugin de IReport y las librerías JasperReport como herramienta de programación, permiten desarrollar Sistemas de Información Geográficos capaces de almacenar información referente a desastres naturales que ocurren en la Provincia de Loja.
- La incorporación de librerías Geotools2 (para gestión de mapas en el lenguaje JAVA) no dio los resultados esperados ya que son muy limitadas, por tratarse de código libre y se encuentran actualmente en desarrollo, por lo que existen procesos relacionados con la manipulación de mapas que no se pudieron programar directamente.
- La metodología ICONIX se adapta fácilmente a proyectos calificados como medianos, como lo es el nuestro, ya que define de manera adecuada las etapas de desarrollo de sistemas; así en nuestro proyecto se pudo identificar de manera minuciosa los requerimientos de usuario debido a que esta metodología tiene bien definida su etapa de análisis, para posteriormente diseñar y construir el sistema pegado a todos los lineamientos de la metodología.
- La realización del presente proyecto nos ha permitido adquirir más conocimientos de las nuevas herramientas y tecnologías para el desarrollo de este tipo de sistemas.
- El trabajo conjunto con los directivos y técnicos del Centro Integrado de Geomática Ambiental (CINFA) que son los usuarios del sistema, nos permitió



obtener nuevos requerimientos que se utilizaron para poco a poco ir perfeccionando la aplicación hasta satisfacer en su totalidad sus necesidades.



10. RECOMENDACIONES

RECOMENDACIONES:

Luego de implantar el sistema podemos dar las siguientes recomendaciones:

- Los proyectos que desarrollen los egresados de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, estén en función de líneas de investigación general con el fin de desarrollar sistemas orientados a resolver problemas de nuestra sociedad.

- Realizar un análisis sobre las herramientas de software, así como el tipo de librerías GIS que se utilizaran para desarrollar una aplicación SIG de escritorio,



en busca de determinar si estas cumplen con los requerimientos necesarios, para así evitar contratiempos a la hora de programar funcionalidades y atributos de los mapas.

- Desarrollar aplicaciones SIG bajo software libre para evitar costos de licencias de software, ya que por lo general suelen ser muy costosas para este tipo de sistemas.
- Utilizar el IDE NetBeans con el lenguaje JAVA para el desarrollo de aplicaciones SIG y específicamente aquellos orientados a resolver problemas relacionados en un entorno geográfico.
- Desarrollar nuevas aplicaciones SIG sobre desastres naturales para las demás Provincias del Ecuador, que sean capaces de inter relacionarse, y compartir información de todo tipo, para de esta manera obtener mayores beneficios.
- Capacitar a los usuarios del sistema y utilizar la aplicación para los fines que fue desarrollada en el Centro Integrado de Geomática Ambiental (CINFA) como medio de consulta e ingreso de datos sobre desastres naturales generados en la Provincia de Loja.



11. BIBLIOGRAFÍA

SITIOS WEB:

- [<http://www.angelfire.com/nt/DesastresNaturales>],[Consulta: 15 Agosto, 2008]
- [<http://www.cegsa.es>] ,[Consulta: 16 Agosto, 2008]
- [http://www.cinfa.edu.ec/Acerca_de_CINFA] ,[Consulta: 17 Agosto, 2008]
- [<http://www.freegis.org>] ,[Consulta: 17 Agosto 2008]
- [<http://www.gabrielortiz.com/Teoria/GIS.asp>] ,[Consulta: 19 Agosto, 2008]
- [<http://www.geotools.org>] ,[Consulta: 5 Septiembre, 2008]
- [<http://www.gnu.org/licenses/licenses.es.html>] ,[Consulta: 2 Septiembre, 2008]
- [<http://www.grass.itc.it>] ,[Consulta: 20 Agosto 2008]
- [<http://www.gvsig.gva.es/framesesp.htm>] ,[Consulta: 19 Agosto, 2008]
- [<http://www.jump-project.org>] ,[Consulta: 25 Agosto, 2008]
- [<http://www.kioskea.net/genie-logiciel/methodes-agiles.php3>] ,[]
- [<http://www.monografias.com/trabajos/fenomenosnatu>],[Consulta: 16 Agosto, 2008]



- [<http://www.monografias.com/trabajos/gis/gis.shtml>] ,[Consulta: 15 Agosto, 2008]
- [<http://www.monografias.com/trabajos12/lsdesast>] ,[Consulta: 26 Agosto, 2008]
- [<http://www.mysql.com>] ,[Consulta: 6 Septiembre, 2008]
- [<http://www.netbeans.org/about/history.html>] ,[Consulta: 10 Septiembre, 2008]
- [<http://www.netbeans.org/community/releases/55>] ,[Consulta: 10 Septiembre, 2008]
- [<http://www.nosolosig.com>] ,[Consulta: 29 Agosto, 2008]
- [<http://www.tdg.lsi.us.es/~csharp>] ,[Consulta: 27 Agosto, 2008]
- [<http://www.wikipedia.org/w/MySQL&redirect>],[Consulta: 15 Septiembre, 2008]
- [http://www.wikipedia.org/wiki/Código_libre] ,[Consulta: 28 Agosto, 2008]
- [http://www.wikipedia.org/wiki/GNU_GPL] ,[Consulta: 26 Agosto, 2008]
- [http://www.wikipedia.org/ Lenguaje _Java],[Consulta:19 Agosto, 2008]
- [<http://www.wikipedia.org/wiki/MySQL>] ,[Consulta: 13 Septiembre, 2008]
- [<http://www.wikipedia.org/wiki/NetBeans>] ,[Consulta: 16 Septiembre, 2008]
- [http://www.wikipedia.org/wiki/Sistema_Inf_Geográf],[Consulta: 20 Agosto, 2008]
- [<http://www.XProgramming.com>],[Consulta: 30 Agosto, 2008]



12. ANEXOS