

# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

## Facultad 6



**TÍTULO:** Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICA**

**Autor:** Sergio Barrera Rodríguez

**Tutor:** Ing. Liset Garcia Padrón

**Co-Tutor:** Ing. Aliana López Costa

**Ciudad de La Habana ,2011**

*"El secreto de cansar a la gente es decirlo todo"*

*Franz Kafka*

## **DEDICATORIA**

*Dedico este trabajo de diploma con mucho cariño a las personas más importantes de mi vida: mis padres, mi hermana, mi abuelo y mi novia, quienes siempre han estado a mi lado en todos los momentos de mi carrera y de mi vida en general, brindándome todo el apoyo y la fuerza que necesito para continuar adelante. Con ellos comparto este triunfo, ya que a ellos se lo debo.*

## **AGRADECIMIENTOS**

*Les agradezco a mis padres Celia Rodríguez Corvea y Sergio Barrera Mena por todo el apoyo brindado durante estos cinco años de carrera, por estar a mi lado en todos los momentos de mi vida ya sean buenos o malos, por todo el apoyo profesional dedicado y por confiar siempre en mí. Los quiero mucho.*

*A mi hermana Dayana Barrera Rodríguez que siempre ha sido un ejemplo para mí a seguir, por estar a mi lado en todos los momentos de mi carrera dándome esa fuerza que siempre me hizo mirar hacia el futuro y continuar adelante a pesar de lo difícil que fuese, Por todo el apoyo profesional que me brindó en esta investigación y por confiar siempre en mí. Eres la mejor del mundo.*

*A mi abuelo que siempre ha estado al tanto de mis cosas, dándome consejos y preocupándose de que siempre sea una mejor persona, por confiar siempre en mí y por todo el apoyo que me brindo a lo largo de estos cinco años.*

*A mi novia Nohary Marín Milian por regalarme estos 6 años de felicidad, por estar junto a mí siempre y consolarme en los momento más tristes y difíciles de mi carrera, por preocuparse siempre por mis cosas como si también fuesen de ella, por ser la mejor novia del mundo y aguantarme todas mis malcriadeces y genios. Mi reina si no fuera por ti ahora mismo no fuera esta persona que soy. Gracias mi amor por todo, tu sabes que te amo más que a mí mismo.*

*A mi Socio y casi como un hermano para mi Yosbel Falero, gracias mi hermano por haber sido siempre mi guía en estos largos cinco años de carrera.*

*A mi primo hermano Abel Delgado Rodríguez por haber estado junto a mí en todos los momentos de mi carrera, brindándome esa ayuda incondicional y por alumbrarme el camino cuando más oscuro estaba.*

*Al resto de mi familia, mis tíos, mis primas lindas ellas saben que las quiero, a los vecinos y amigos de mi pueblo.*

*A mis grandes amigos: Franier, Adrian, Yadriel, Figueras, Chapman, Rene, Osmel, Rodolfo, Diosbel, Eduanys, y David.*

## ***DECLARACIÓN DE AUDITORÍA***

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente del mismo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de junio del 2011.

---

Sergio Barrera Rodríguez

Autor

---

Ing. Liset Garcia Padrón

Tutor

## ***RESUMEN***

En la Universidad de las Ciencias Informáticas se desarrollan proyectos productivos, como parte de la realización de estos proyectos un grupo de trabajo de la facultad 6 está desarrollando un SIG que lleva por nombre GeneSIG. Este es un Sistema de Información Geográfica que está desarrollado con herramientas y tecnologías libres

La plataforma GeneSIG conlleva a guardar un gran número de información así como una lista ordenada de los mapas contenidos en ella, y se vio la necesidad de incorporarle un módulo que facilitara este proceso. Así surgió el catálogo de mapas LiberMaps, este permite realizar la edición dinámica de los objetos de un mapa, de modo que el usuario pueda interactuar más fácil y organizadamente. En el mismo, como en todo catálogo de mapas, se trabaja con grandes cantidades de información geográfica por lo que se observó la necesidad de documentar toda esta información y facilitar una mejor organización y localización de los datos. Así surge el Sistema para la Gestión de Metadatos Geográficos.

Se observó la necesidad de agregarle un módulo a este sistema que permita registrar toda la información existente de personas u organizaciones que realizan estos metadatos geográficos y así acudir a ellos en caso de necesidad. El sistema está desarrollado en un ambiente web con tecnologías libres en su mayoría. Teniendo en cuenta la metodología seleccionada: RUP, se documentó todo el proceso generando los artefactos correspondientes a cada flujo de trabajo.

## ***PALABRAS CLAVES***

Ambiente web, Gestión, Metadatos, Sistema informático, Tecnologías libres.

## ***INDICE***

INTRODUCCION .....	14
<b>CAPITULO 1: Fundamentación Teórica.....</b>	<b>19</b>
1.1. Introducción .....	19
1.2. Conceptos que ayudarán a entender el problema .....	19
1.2.1. Contacto.....	19
1.2.2. Agenda de contactos.....	19
1.2.3. Metadato .....	20
1.2.4. Metadato geográfico.....	20
1.2.5. Sistema de información geográfica .....	20
1.2.6. Información .....	20
1.3. Estado del Arte.....	21
1.3.1. Utilización de las agendas de contactos en diversos campos .....	21
1.3.1.1 Ejemplo de Agendas de contactos .....	21
1.4. Soluciones Existentes .....	23
1.4.1. Sistemas que utilizan Módulo de Agendas de Contactos en el Mundo Vinculados a Metadatos. ....	23
1.5. Situación Problemática.....	24
1.6. Conclusiones .....	25
<b>CAPITULO 2: Tendencias y tecnologías a utilizar. ....</b>	<b>26</b>



2.1 Introducción .....	26
2.2 Metodología de Desarrollo de Software.....	26
2.3 Lenguaje Unificado de Modelado (UML). ....	27
2.4 Servidores Web.....	28
2.4.1 Servidor HTTP Apache. ....	28
2.5 Lenguajes de Programación Web. ....	28
2.5.1 Lenguajes del lado del servidor.....	29
2.5.2 Programación del lado del cliente.....	30
2.6 Sistemas Gestores de Bases de Datos. ....	31
2.7 Herramientas CASE.....	32
2.8 Framework.....	32
2.8.1 Framework Symfony .....	33
2.9.1 NetBeans .....	34
2.10 Conclusiones.....	35
<b>CAPÍTULO 3: Presentación de la solución propuesta.....</b>	<b>36</b>
3.1. Introducción .....	36
3.2 Modelo de dominio.....	36
3.2.1. Diagrama de clases del Modelo de Dominio.....	36
3.2.2. Glosario de Términos del Dominio .....	37
3.3.Modelo del Sistema	
.....	37
3.3.1. Requisitos Funcionales .....	37

3.3.3. Descripción del Sistema Propuesto.....	41
3.3.3.1 Descripción de los actores. ....	41
3.3.4 Descripción de los Casos de Uso del Sistema.....	42
3.3.4.1 Descripción Textual del Caso de Uso Gestionar datos de Persona.....	42
<b>CAPÍTULO 4: Construcción de la solución propuesta.....</b>	<b>47</b>
4.1. Introducción. ....	47
4.2. Diseño.....	47
4.2.1. Descripción general del funcionamiento del Symfony y ExtJs. ....	47
4.3. Patrones de Diseño.....	49
4.3.1. Patrones GRASP implementados en Symfony.....	49
4.3.2. Tipos de patrones GOF que implementa Symfony. ....	50
4.3.3. Patrones implementados en la vista de la aplicación.....	51
4.4. Diagramas de Clases del Diseño. ....	51
4.4.1. Realización del Caso de Uso: Gestionar Registro de Contacto .....	52
4.5. Diseño de la Base de Datos. ....	52
4.5.1. Diagrama de Clases Persistentes. ....	53
4.5.2. Modelo Entidad-Relación. ....	54
4.6. Modelo de Despliegue.....	56
4.7. Modelo de Implementación .....	56
4.8. Conclusiones.....	59
4.9. Validación de la solución propuesta. ....	60
4.9.1. Caso de prueba del Casos de uso Gestionar registro de Contacto. ....	60

4.9.2. Caso de prueba del Casos de uso Gestionar datos de Persona.....	62
CONCLUSIONES .....	65
REFERENCIA BIBLIOGRÁFICA.....	67
BIBLIOGRAFIA.....	70
ANEXOS.....	71
Anexo 1.....	71
Anexo 2.....	79

## ***INDICE DE FIGURAS Y TABLAS***

### ***FIGURAS***

Figura 1: Disciplinas, Fases e Iteraciones de la Metodología RUP.....	27
Figura 2: Modelo de Dominio .....	37
Figura 3: Caso de Uso del Sistema.....	42
Figura 4: Estructura del Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos. ....	49
Figura 5: Diagrama de Clases del Diseño CU Gestionar Registro de Contacto.....	52
Figura 6: Diagrama de Clases Persistentes. Paquete Información del Módulo para la Agenda de Contactos.....	53
Figura 7: Diagrama de Clases Persistentes. Paquete Información del Módulo para la Agenda de Contactos.....	53
Figura 8: Diagrama de Clases Persistentes. Paquete Información del Módulo para la Agenda de Contactos.....	54
Figura 9: Modelo Entidad Relación. Paquete Información del Módulo para la Agenda de Contactos.....	55
Figura 10: Modelo Entidad Relación. Paquete Información del Módulo para la Agenda de Contactos.....	55
Figura 11: Modelo Entidad Relación. Paquete Información del Módulo para la Agenda de Contactos.....	55
Figura 12: Diagrama de Despliegue.....	56
Figura 13: Diagrama de Componentes. Vista General. ....	57
Figura 14: Diagrama de Componentes. Paquete Modelo.....	57
Figura 15: Diagrama de Componentes. Paquete Ext. ....	58

Figura 16: Diagrama de Componentes. Paquete Módulos. ....	58
Figura 17: Diagrama de Componentes. Paquete Módulo. ....	59

## ***TABLAS***

Tabla 1: Descripción Textual del Caso de Uso Gestionar datos de Persona. ....	44
Tabla 2: Gestionar Datos de Persona .....	46
Tabla 3: Sección a probar en el Caso de Uso .....	61
Tabla 4: Descripción de Variables.....	61
Tabla 5: Matriz de Datos .....	62

## ***INTRODUCCION***

El desarrollo del hombre en la sociedad ha inquirido en la necesidad de cambios sucesivos aparejados a las conductas de transmitir y preservar la información. Actualmente las prácticas gerenciales han establecido en conjunto con el desarrollo de las nuevas tecnologías de información y comunicación, herramientas eficaces en el proceso de gestión de la información, que patentizan el capital acumulado de datos dentro de las diferentes esferas sociales y la búsqueda de nuevos proyectos en pos de desarrollar los sistemas de registros que hoy representan grandes volúmenes de información. La sociedad actual ha fijado sistemas de información encauzados al desempeño organizacional y al perfeccionamiento de las prácticas de gestión informacional donde los grandes sistemas de gestión en términos de calidad y satisfacción del usuario juegan un papel significativo.

Los Sistemas de Información Geográfica (SIG) son un ejemplo del avance en las tecnologías de la información, ya que estos contienen un buen número de datos que ayudan a facilitar el trabajo del individuo. Un SIG es una herramienta para superar la visión sectorial y consolidar una comprensión integral del territorio, mediante la interacción de las dimensiones ambientales, culturales, económicas y sociales. Estos desempeñan un papel fundamental en la representación y análisis de la información geográfica, debido a que proveen los medios necesarios para la captura, organización, manipulación y uso de la información.

El avance en el desarrollo de los SIG junto a los logros en otros campos de la ciencia han hecho que la complejidad, calidad y diversidad de los datos aumenten aceleradamente, por lo que fue necesario crear metadatos que permitieran agrupar todos estos elementos de forma tal que se lograra un manejo eficiente de los datos. Se hacía necesario contar con una documentación organizada de la información existente y de la que se creara posteriormente con el objetivo de facilitar las búsquedas de esta información, poder caracterizarla y analizarla sin mucha dificultad. La solución a este problema la constituyen los metadatos. Los metadatos geográficos no son más que una descripción de datos geoespaciales específicos que permite distinguirlos a través de aspectos como son: calidad, actualización referencia geoespacial autor, entre otros.

Con la proliferación de las redes sociales se han establecido variadas plataformas que fomentan la colaboración y la aplicación de herramientas que permiten crear canales de comunicación e interacción entre las personas con algún tipo de interés o necesidad específica. La agenda de contactos es un ejemplo más de las herramientas creadas para registrar información de personas u organizaciones que posibilitan tener almacenado datos de interés.

Cuba también se está adentrando en el mundo tecnológico actual, como ejemplo de ello está la creación de la Universidad de las Ciencias Informáticas (UCI). En esta se desarrollan proyectos productivos que permiten a los estudiantes vincular lo aprendido en clases con la producción y al mismo tiempo ayudar a informatizar varios sectores de la sociedad y a contribuir con países hermanos. Como parte de la realización de estos proyectos un grupo de trabajo de la facultad 6 de la UCI está desarrollando en un SIG que lleva por nombre GeneSIG. Este es un Sistema de Información Geográfica que está desarrollada con herramientas y tecnologías libres y en entorno Web. La misma cuenta con un gran grupo de funciones que posibilitan al usuario una mayor ubicación en todo el país sin necesidad de consultar mapas de papel, como son:

Ubicación de centros turísticos del país, zonas fértiles para el cultivo y precisar zonas de posible inundación.

Este grupo de servicios que brinda la plataforma GeneSIG como SIG conlleva a guardar un gran número de información así como una lista ordenada o clasificada de los mapas contenidos en ella y por ello se vio la necesidad de incorporarle un módulo que facilitara este proceso. Así surgió el catálogo de mapas LiberMaps. Este catálogo se encarga de la seguridad de los datos y funcionalidades, permite realizar la edición dinámica de los objetos de un mapa y garantiza una estructura más organizada, de modo que el usuario pueda interactuar con ellos más fácil y organizadamente. En el mismo, como en todo catálogo de mapas, se trabaja con grandes cantidades de información geográfica por lo que se observó la necesidad de documentar toda esta información y facilitar una mejor organización y localización de los datos. Era preciso mantener el acervo del conjunto de datos contenidos en el mismo y proporcionar la información necesaria para interpretar y procesar los datos de éste.

Con el transcurso del trabajo se observa la necesidad de agregarle módulos a este sistema para facilitar y mejorar su uso. Se realizó una investigación profunda sobre el

tema y como resultado se planteó agregarle a dicho sistema una herramienta que registrara toda la información existente de personas u organizaciones que realizaran estos metadatos, esta herramienta se le denominará Agenda de Contactos.

Por todo lo anterior expuesto se decidió realizar la investigación identificando como **problema a resolver** ¿Cómo mantener y actualizar los datos de las personas u organizaciones que poseen relación con la información geográfica contenida en el Sistema para la Gestión de Metadatos Geográficos, de forma que puedan ser contactadas en caso de necesidad?

A partir del problema anterior se define entonces como **objeto de estudio** los Procesos de gestión de información de contactos.

Se determina como **objetivo general** de la investigación Desarrollar un módulo para el Sistema para la Gestión de Metadatos Geográficos que permita la gestión de la información de sus contactos.

El **campo de acción** lo constituye la Automatización de los procesos de gestión de información de contactos para el Sistema de Gestión de Metadatos Geográficos.

Teniendo en cuenta todo lo anterior se plantea como **idea a defender** Contar con un módulo que gestione la información de contactos que poseen relación con la información geográfica contenida en el Sistema para la Gestión de Metadatos Geográficos, permitirá acceder a la misma de manera más eficiente y rápida.

Para dar cumplimiento al objetivo de esta investigación se definieron las siguientes **tareas investigativas**:

- Caracterizar el proceso de gestión de información de contactos para el Sistema para la Gestión de Metadatos Geográficos.
- Caracterizar las tendencias y tecnologías actuales a desarrollar.
- Identificar requerimientos funcionales y no funcionales para desarrollar el Módulo para la gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos.
- Diseñar el Módulo para la gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos.



- Implementar el Módulo para la gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos.
- Documentar la investigación con vistas a su socialización.

Para desarrollar estas tareas se utilizarán los siguientes **métodos científicos**:

#### **Métodos Teóricos:**

**Análisis histórico – lógico** para el análisis histórico de los procesos de gestión de información de contactos y de las tendencias actuales en el desarrollo de módulos de este tipo.

**Análisis Síntesis:** Para realizar una revisión bibliográfica de todos los temas asociados a la investigación y luego hacer una recopilación de la bibliografía a utilizar.

**Modelación** para realizar diagramas y representar todo el proceso de la gestión de información de contactos para el Sistema de Gestión de Metadatos Geográficos facilitando un mejor entendimiento de la solución a implementar.

#### **Métodos Empíricos:**

**Observación** utilizada durante todo el proceso de investigación, para identificar el dominio del problema.

Se espera como **Posibles resultados** realizar un Módulo que gestione la información de contactos que poseen relación con la información geográfica contenida en el Sistema para la Gestión de Metadatos Geográficos.

La investigación estará estructurada de la siguiente manera:

**Capítulo 1. Fundamentación Teórica:** En el mismo se hace referencia a conceptos asociados al dominio del problema así como un estudio del estado del arte y las soluciones existentes en el mundo.

**Capítulo 2. Tendencias y tecnologías actuales a utilizar:** Se realiza un análisis de las tecnologías y metodologías para la construcción de software existentes en la actualidad y se reflejan las que se van a utilizar en el sistema a implementar.

**Capítulo 3. Presentación de la solución propuesta:** Se analizan los procesos de gestión de información de contactos para describir la propuesta del sistema. Se determinan los

conceptos principales para realizar el modelo de dominio y se identifican los requisitos funcionales y no funcionales del sistema representándose en el diagrama de casos de uso del sistema.

**Capítulo 4. Construcción de la solución propuesta:** Se modela el diseño del sistema y se plantea el modelo de datos. Se efectúa la implementación del sistema, reflejándose los diagramas correspondientes como son el de componentes y el de despliegue.

## ***CAPÍTULO 1: Fundamentación Teórica.***

### **1.1. Introducción**

En este capítulo se abordarán distintos temas que permiten justificar teóricamente la investigación, entre ellos los conceptos asociados al dominio del problema como qué es un contacto y más específico qué es una agenda de contactos, qué es un metadato y a su vez un metadato geográfico, qué es un SIG y otros. Se indagaran además, los modelos internacionales que existen para la creación de agendas de contactos, así como lo referente a éstas en Cuba. Se brindará una explicación de la problemática que llevó a esta investigación y se hace un estudio de los sistemas que utilizan un módulo de agendas de contactos en el mundo.

### **1.2. Conceptos que ayudarán a entender el problema**

De forma tal que se pueda entender mejor los principales aspectos que conforman esta investigación es significativo conocer algunos conceptos relacionados al dominio del problema

#### **1.2.1. Contacto**

Un contacto es una persona u organización que se le registra sus datos ya sean personales o no, para poder dirigirnos a él en caso de necesidad. Este quedaría registrado en algún medio ya sea una agenda o simplemente un papel y a partir de ese momento pasaría a formar parte de contacto.

#### **1.2.2. Agenda de contactos**

Una agenda de contacto es un libro o cuaderno que se utiliza para apuntar los datos de personas u organizaciones que son de gran interés para cierta persona o empresa y así en caso de necesitar localizarlos resulte sencillo. **(1)**

Los datos pueden ser estos por ejemplo:

- Nombre
- Apellidos
- Carnet de identidad
- Dirección de vivienda
- Dirección del trabajo
- Número de teléfono

### 1.2.3. Metadato

Un metadato llamado por primera vez en 1960 por “Datos de datos”, es toda aquella información descriptiva sobre el contexto, calidad, condición o característica de un recurso, datos u objeto que tiene la finalidad de facilitar su recuperación, autenticación y evaluación. **(2)**

### 1.2.4. Metadato geográfico

Un metadato es una información que describe entre otras, la calidad, distribución, actualidad y referencia espacial de un conjunto de datos. Muchos especialistas llaman a los metadatos como “datos acerca de los datos”. Un metadato geográfico trata de caracterizar los datos espaciales respondiendo al quién, qué, cuándo, dónde, por qué y cómo de estos datos. **(3)**

Los metadatos para que sean realmente importantes deben estar compilados en bases de datos y disponibles en Servicios de Consulta de Información Geográfica o ClearingHouse<sup>1</sup>

### 1.2.5. Sistema de información geográfica

El profesor David Rhind (1989) plantea que un **Sistema de Información Geográfica (SIG)** es un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados (georreferenciados), para la solución de los problemas complejos del manejo y planeamiento territorial. **(4)**.

### 1.2.6. Información

Todo aquello que permite adquirir cualquier tipo de conocimiento se denomina información. Existe información cuando se revela algo que hasta ahora era desconocido; la función de la información es aumentar el conocimiento del receptor o reducir su incertidumbre. Para lograr una información son indispensables los datos que una vez procesados, es decir ordenados y clasificados, constituyan una información útil y funcional. En toda organización es necesaria la información que surge como consecuencia de las actividades que se llevarán a cabo para alcanzar sus objetivos.

---

<sup>1</sup> Un ClearingHouse es una red distribuida de productores y usuarios de datos espaciales que permite encontrar y acceder a metadatos y datos geográficos, con escasas restricciones.

### 1.3. Estado del Arte

#### 1.3.1. Utilización de las agendas de contactos en diversos campos

Las Agendas de contacto en el mundo son muy usadas en diferentes esferas por su gran importancia y capacidad de brindar una organización total del tiempo de trabajo y de tener a todas las personas que proporcionan interés localizado para en caso de necesitar de ellas, contactar rápidamente.

La telefonía móvil fue una de las primeras en darse cuenta que las agendas de contactos son de gran utilidad en la vida diaria por la serie de servicios que esta brinda. Es por eso que tomaron la alternativa de incorporarle a los teléfonos móviles servicios de agendas de contacto, esto mejoraría el uso del teléfono y brindaría a los usuarios una mejor organización de sus contactos, así como tener registrados todos los datos personales de éstos. El sistema operativo Windows también trae incluida en su interior una agenda de contacto muy bien planificada ya que le permite al usuario hacer varias acciones de gran comodidad y organización. Este admite que se puedan guardar todo tipo de datos de personas como teléfono, dirección de vivienda, dirección de correo, entre otros.

##### 1.3.1.1 Ejemplo de Agendas de contactos

###### **Open Contacts**

Open Contacts es una libreta de direcciones o agenda de contactos, el propósito de esta utilidad es el mismo igual. De carácter gratuito y diseñado sólo para Windows, da un completo sistema para gestionar todo lo que tenga que ver con relaciones sociales, con contactos.

El programa es bastante personalizable y además flexible. **Open Contacts** puede importar datos desde programas como Outlook, vCard, Excel, Eudora y otros. También destaca por ser de fácil manejo e interactivo con el usuario en todas sus operaciones. (5)

###### **ContactKeeper 1.4.3**

1. ContactKeeper es una práctica agenda personal que mantiene perfectamente organizados y siempre a mano todos los datos de amigos, compañeros de trabajo, clientes y familiares.

2. Crear entradas es fácil y rápido. Además, en cada entrada se puede añadir varias direcciones de correo electrónico y múltiples números de teléfono, muy útil para incluir los datos personales y del trabajo.
3. Como extra, cada entrada incluye un apartado para notas de cualquier índole (con soporte para enlaces a páginas web) y un campo de fecha de nacimiento. Si completas esta fecha, ContactKeeper generará automáticamente un calendario de cumpleaños y avisará varios días antes del evento, así te dará tiempo a comprar el regalo. **(6)**
4. Uno de los campos que brinda son los siguientes:
  - Nombre
  - Apellido
  - Fecha de cumpleaños
  - Domicilio
  - Teléfono
  - E-mail

#### **Agenda XP 2.5.2**

Múltiple agenda de contactos que se rige única y exclusivamente en los datos básicos de contactos, como son el nombre, dirección y teléfono.

Es una agenda extremadamente sencilla y fácil de usar, con los requisitos mínimos de cualquier agenda, pero la interfaz es agradable y en caso de sólo necesitar eso, una simple agenda, cumple perfectamente.

Dispone de la opción de poder tener tantas agendas diferentes como se desee, cada una de ellas protegida por contraseña. **(7)**.

#### **MyPortablePIM 1.3.0**

MyPortablePIM es una agenda que puede ejecutar desde una memoria USB o cualquier dispositivo extraíble. Cuenta con calendario, lista de tareas, notas rápidas, agenda de contactos y alarmas. Son funciones que se integran muy bien entre ellas y son rápidas de manejar.

Algo más de configuración requieren los demás módulos de myPortablePIM, como el lector de correo GMail, la lanzadera de programas, el lector de noticias RSS y el panel de previsiones climatológicas. Cada uno de estos paneles se puede mover de sitio para que compongas la ventana de myPortablePIM como quieras.

Parece que el autor de myPortablePIM ha preferido amasar la mayor cantidad de funciones posible, sin importar demasiado su utilidad o potencia. Por suerte, las pestañas se pueden cerrar para que myPortablePIM tenga un aspecto más pulcro y manejable. Eso no parece reducir el consumo de memoria, bastante elevado.

A pesar de sus defectos, myPortablePIM es un organizador personal más que recomendable, especialmente si se trabaja con iCal y GMail con regularidad. Eso sí, se debería ejecutar en ordenadores con bastante memoria. **(8)**.

### **Agenda MSD Multiusuario 11.0**

Agenda MSD Multiusuario es un gestor de información personal y profesional muy completo y fácil de usar que incluye módulos de calendario, agenda telefónica, avisos, tareas, diario, fichas de datos, patrimonio, presupuesto, salud y música.

Esta versión Multiusuario de Agenda MSD permite compartir los datos de la agenda a través de una red local o incluso a través de Internet, además de permitir disponer de datos privados en el ordenador, incrustar documentos de todo tipo en los registros, llamar por teléfono a los contactos, visitar sus páginas Web o enviarles correo electrónico directamente.

Incluye el programa Administrador de Agenda MSD, que deberá ser instalado en el ordenador que hará de servidor y en él incluirá el archivo de instalación de la versión cliente de Agenda MSD Multiusuario que se instalará por los usuarios. También incluye un completo módulo de mensajería en red local. **(9)**.

## **1.4. Soluciones Existentes**

### **1.4.1. Sistemas que utilizan Módulo de Agendas de Contactos en el Mundo Vinculados a Metadatos.**

CatMDEdit es una herramienta de edición de metadatos que facilita la documentación de los recursos, con especial atención a la descripción de los recursos de información

geográfica. Se trata de una iniciativa del Instituto Geográfico Nacional de España (IGN), que es el resultado de la colaboración científica y técnica entre el IGN y el Grupo de Sistemas de Información Avanzados (IAAA) de la Universidad de Zaragoza con el apoyo técnico de GeoSpatiumLab (GSL)

Herramientas adicionales para facilitar la edición de metadatos:

- Contactos herramienta de repositorio. Permite la reutilización de la información de contacto (por ejemplo, nombre, dirección, teléfono) de las organizaciones e individuos, que deben ser llenados en varios elementos de metadatos. Gracias a este componente, la información de contacto de una persona responsable se inserta una sola vez y utilizar siempre que sea necesario **(10)**.

Este cuenta con un Módulo para la agenda de contacto muy bien implementado, por eso se desea de cierta manera registrarse por este excelente programa a la hora de darle solución a esta investigación.

### **1.5. Situación Problemática**

Las agendas de contacto surgen con la necesidad de tener una localización amplia de personas u organizaciones que son de gran importancia para cierta entidad o persona, éstas son muy utilizadas en diversas empresas que por alineación y mejoramiento del trabajo tienen a todos sus trabajadores u organizaciones localizados para determinada situación.

En la Universidad de las Ciencias Informáticas (UCI) se desarrolla un proyecto productivo en la facultad 6 donde pertenecen un grupo de estudiantes y profesores, en el mismo se trabaja en la plataforma GeneSIG. Este es un Sistema de Información Geográfica (SIG), al que se le incluyó un módulo para la gestión de metadatos geográficos con la idea de organizar mejor la información y además tenerla más simplificada ya que en su inicio se trabajaba con grandes volúmenes de información y así se dificultaba el proceso. Más adelante con investigaciones realizadas por parte de la dirección del proyecto se advierte la necesidad de registrar de alguna forma todos aquellos datos de personas u organizaciones creadores de estos metadatos y a partir de ahí surge la idea de incorporarle a este módulo para la gestión de metadatos geográficos, una herramienta capaz de guardar toda esa información de dichas personas u organizaciones para en



caso de necesitarlos poder contactarlos fácilmente. A esta herramienta se le denomina Agenda de Contactos para Metadatos Geográficos.

### **1.6. Conclusiones**

En la elaboración de este capítulo se abordaron conceptos generales que sirvieron de guía para lograr un mejor entendimiento de esta investigación. Se tomó como decisión registrarse sobre la herramienta CatMDEdit para una mejor conexión del módulo.

## ***CAPÍTULO 2: Tendencias y tecnologías a utilizar.***

### **2.1 Introducción**

En el presente capítulo se realiza un estudio de las distintas metodologías de desarrollo de software a utilizar que ayudará a entender mejor el funcionamiento de todo el trabajo. Se hace una caracterización de tecnologías y herramientas que se utilizan en distintos software existentes en la actualidad. Se establece una comparación entre alguna de ellas tomando como conclusión cuál se utilizará para la elaboración del sistema propuesto.

### **2.2 Metodología de Desarrollo de Software.**

Las metodologías y estándares utilizados en el desarrollo de software proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación de la aplicación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado.

Existen dos grupos de metodologías de desarrollo de software: Metodologías Ágiles y Metodologías Tradicionales o Robustas. Entre las metodologías más utilizadas de los últimos tiempos se encuentra Rational Unified Process (RUP) **(11)**.

RUP posee numerosas ventajas pues se centra en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodología es más eficaz y necesaria y además determina las etapas y la documentación que es necesaria a realizar durante el proyecto de creación del software.

También posee tres aspectos que la hace única los cuales se resumen en tres frases claves: dirigido por casos de usos, centrado en la arquitectura, e iterativo e incremental. El Proceso Unificado de Desarrollo es una solución al problema del software. **(12)**.

Según RUP, el desarrollo del software se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición, cada una de las cuales es desarrollada mediante ciclos de iteraciones.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales como se muestra en la figura 6. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. **(11)**

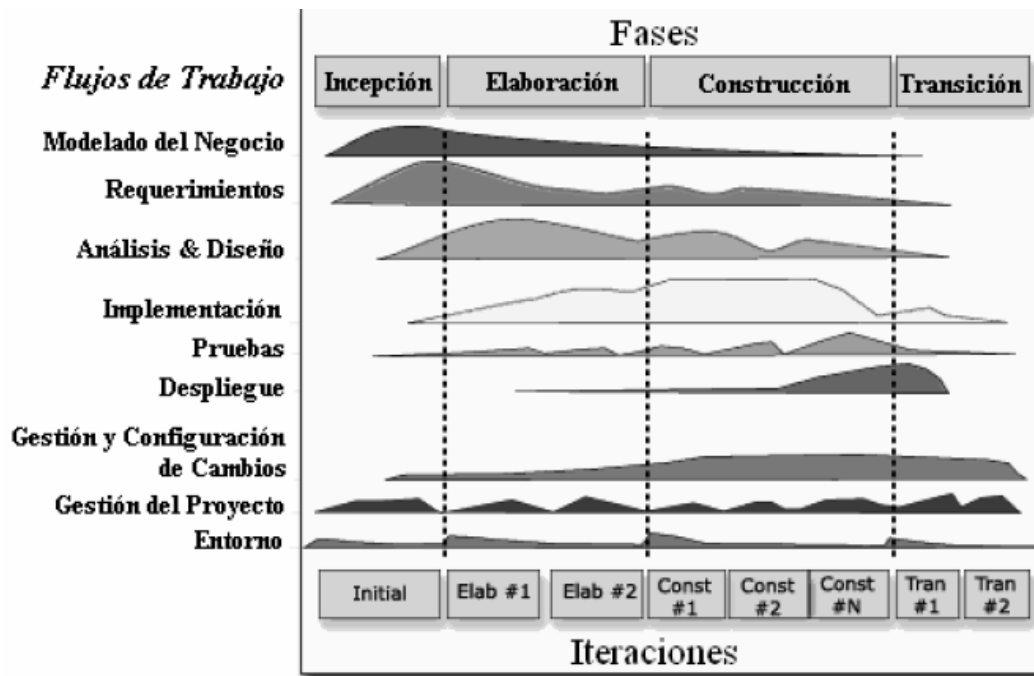


Figura 1: Disciplinas, Fases e Iteraciones de la Metodología RUP.

Por los argumentos planteados se determina como metodología de desarrollo de software a utilizar en la implementación del módulo de La Agenda de Contactos para el Sistema de Metadatos Geográficos, debido a las ventajas descritas.

### 2.3 Lenguaje Unificado de Modelado (UML).

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de lo que estos diagramas y símbolos significan. (27).

Sus principales funciones son las de Visualizar, Especificar, Construir y Documentar. A través de este lenguaje se puede modelar, lo mismo sistemas de software, hardware u organizaciones del mundo real. Un modelo UML está compuesto por tres clases de bloques de construcción: Elementos, Relaciones y Diagramas (13).

Debido a las características antes mencionadas, se escoge este lenguaje de modelado ya que permite construir modelos por ingeniería inversa a partir de sistemas existentes. UML es fácil de aprender y utilizar, con la modelación de los de artefactos durante las primeras fases de ciclo de vida del software, se posibilita que en fases posteriores los implementadores tengan un mayor dominio y una mejor comprensión sobre qué es lo que se debe implementar, permitiendo tanto al cliente como a los desarrolladores tener una

representación real de los alcances y la factibilidad que puede o no llegar a tener el producto **(14)**.

## **2.4 Servidores Web.**

Un servidor Web es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Lenguaje). La comunicación de estos datos entre cliente y servidor se hace por medio de un protocolo, concretamente del protocolo HTTP. El servidor es el responsable de proporcionar el acceso a los recursos solicitados que están bajo el control del sistema operativo. Estos recursos pueden ser estáticos, como páginas HTML o texto y dinámicos, como por ejemplo CGI's. (Common Gateway Interface o Pasarela de Interface Común) **(15)**.

### **2.4.1 Servidor HTTP Apache.**

Se seleccionó como servidor web El Servidor HTTP Apache por las siguientes ventajas: Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos. Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Es un servidor altamente configurable de diseño modular ya que es muy sencillo ampliar las capacidades del mismo. Trabaja con gran cantidad de lenguajes como Perl, PHP y otros de script. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor, esto es posible al configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Presenta un alto nivel de configuración en la creación y gestión de logs pues facilita la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor **(16)**.

El servidor Web elegido es Apache, pues brinda varias ventajas dentro de las que se encuentra fundamentalmente que es de código abierto, otros aspectos muy importantes que lo caracterizan son su estructura modular, además de ser multiplataforma, extensible y la facilidad de encontrar ayuda y soporte para el mismo.

## **2.5 Lenguajes de Programación Web.**

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de términos monosémicos (es decir, con sentido único) y una regla principal que resume las demás, todas estas son utilizadas por el programador a través de

las cuales crea un programa o subprograma. (Enciclopedia Libre Universal en Español)  
En este epígrafe se abordan algunas características de lenguajes de programación del lado del servidor y del lado del cliente, para poder argumentar con estos elementos los lenguajes seleccionados.

### **2.5.1 Lenguajes del lado del servidor.**

#### **PHP**

PHP (acrónimo de PHP: Hipertexto Preprocessor), es un lenguaje de programación interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor, diseñado originalmente para la creación de páginas Web dinámicas. Es de gran velocidad por lo que no requiere de recursos del sistema y además se integra perfectamente con servidores. Es libre y está disponible bajo la licencia GPL<sup>2</sup>, es multiplataforma por lo que no tendrá ningún inconveniente al usarlo en cualquier computadora de la Universidad. Se caracteriza por la simplicidad de su código y por la amplia documentación que brinda. **(17)**.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase MySQL, Informix, entre otras.

- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.

---

<sup>2</sup> Licencia Pública General de GNU.

- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se logra hacer cualquier cosa que se pueda realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas **(18)**.

Se seleccionó como lenguaje de programación PHP, por su rapidez, su facilidad de aprendizaje, su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos y el hecho de que se distribuye de forma gratuita bajo una licencia de código abierto.

### **2.5.2 Programación del lado del cliente.**

Se seleccionó como lenguaje del lado del cliente ExtJS. Este es un framework JavaScript para interfaces avanzadas. Con el tiempo y la experiencia, se ha mejorado y desarrollado, integrándole librerías para infinidad de problemas al programar en la web. Entre las principales actividades que se hacen se encuentran: mensajes de advertencia, listas desplegables, validación de campos en formularios, entre otras. Con su uso, se hace más fácil la programación.

Javascript es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario **(19)**.

El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". Ajax se trata de varias tecnologías independientes que se unen de forma novedosa. Las tecnologías que forman AJAX son: XHTML y CSS, para crear una presentación basada en estándares; DOM, para la interacción y manipulación dinámica de la presentación; XML, XSLT y JSON, para el intercambio y la manipulación de información; XMLHttpRequest, para el intercambio asíncrono de información; JavaScript, para unir todas las demás tecnologías. **(20)**.

ExtJS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, ExtJS no se preocupará de lo que pase en el

servidor. Hay docenas de widgets a escoger en ExtJS, incluyendo composiciones automáticas de páginas, pestañas, menús, barras de herramientas, diálogos, vistas en árbol. Proporciona un selector de nodos DOM extremadamente poderoso llamado DomQuery (puede usarse como una librería independiente, pero en el contexto de ExtJS se usará para seleccionar elementos para poder interactuar con ellos a través de la interfaz Element; contiene muchos de los métodos y propiedades de DOM que se necesitarán, proporcionando una interfaz conveniente, unificada y multinavegador) **(19)**.

Del lado del cliente se propone utilizar el framework ExtJs para el desarrollo de las interfaces pues además de permitir la reutilización de código, es independiente o adaptable a frameworks diferentes y proporciona una interfaz conveniente, unificada y multinavegador.

## **2.6 Sistemas Gestores de Bases de Datos.**

Como sistema gestor de base de datos se ha elegido PostgreSQL, por las características que se detallan más adelante.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD4 y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

A continuación se exponen algunas de las características más importantes de PostgreSQL:

### Generales

- Completa documentación
- Licencia BSD
- Funciona en la mayoría de los Sistemas Operativos más utilizados, incluyendo Linux y Windows.
- Utiliza principalmente como lenguaje de consulta a la base de datos SQL.
- Admite varios lenguajes procedurales como Java, Perl, Python, Ruby, Tcl, C/C++, así como su lenguaje nativo (PL/PGSQL) que es muy similar al PL/SQL de Oracle.

## 2.7 Herramientas CASE.

Como herramienta de modelado se utilizará Visual Paradigm for UML (Lenguaje de modelado Unificado), por su estabilidad de ejecución en diferentes sistemas operativos y la facilidad de abrir y trabajar con un modelo UML utilizando el mismo programa sin importar el sistema operativo y sin afectar en absoluto el trabajo hecho; además destacar que esta herramienta guarda todo el modelo en un solo fichero.

### Características

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio.
- Documento Modelado colaborativo con CVS y Subversión.
- Interoperabilidad con modelos UML2.
- Código a modelo.
- Código a diagrama.
- Ingeniería inversa Java, C++.
- Generación de código.
- Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso.
- Diagramas de flujo de datos.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas **(21)**.

Además se seleccionó Visual Paradigm en primer lugar porque es una herramienta que puede ser ejecutada sobre entornos libres, de la cual se ha ido adoptando su utilización en la UCI precisamente por esta razón. Además, tiene una buena integración con IDEs, es muy personalizable y soporta UML 2.1 y lenguajes de programación.

## 2.8 Framework.

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas **(22)**.



### 2.8.1 Framework Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

#### Características de Symfony:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “*convenir en vez de configurar*”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de *mejores prácticas* y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Este framework está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows. Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. **(22)**.

Se escogió Symfony, ya que permite un rápido desarrollo de aplicaciones web construidas con técnicas probadas y estándares, el trabajo es más sencillo ya que no necesita tanto código, además de tener tratamiento de errores.

Las aplicaciones Web resultantes son más sencillas de mantener y/o ampliar debido al uso del patrón Modelo Vista Controlador (MVC), que divide la aplicación en capas separadas y por último, pero no menos importante, es que es publicado bajo licencia de código abierto.

## **2.9 Entorno Integrado de Desarrollo (IDE).**

Un entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse a un sólo lenguaje de programación o a varios.

### **2.9.1 NetBeans**

El IDE NetBeans es 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. Es un producto libre y gratuito sin restricciones de uso.

NetBeans es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. Además soporta el desarrollo de todos los tipos de aplicación Java. Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactoring.

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 **(23)**.

Se ha seleccionado la herramienta NetBeans como editor de php, pues ofrece un eficiente auto-completado de código, marcado de error de PHP, plantillas, macros, control de versiones, también es muy importante decir que es multiplataforma y de código abierto lo que se corresponde con el objetivo principal de este proyecto.

## **2.10 Conclusiones.**

En este capítulo se ha realizado un estudio profundo de las tecnologías y herramientas, de esta forma se ha fundamentado la selección de dichas herramientas para el desarrollo de la solución propuesta. Se seleccionaron los framework ExtJs del lado del cliente y Symfony del lado del servidor, en los cuales se utilizan como lenguaje de programación, Javascript y php respectivamente. Para llevar a cabo la implementación del sistema se seleccionó el IDE NetBeans en su versión 6.9, el sistema de gestión de base de datos PostgreSQL en su versión 8.2 y Apache como servidor Web. La metodología que guiará el proceso de desarrollo será RUP que emplea como lenguaje de modelado UML y se utilizará la herramienta CASE Visual Paradigm. El correcto empleo de estas tecnologías y herramientas contribuye a obtener un software con calidad. Después de haber hecho una completa selección de las tecnologías y herramientas para desarrollar el presente proyecto, se está en condiciones de realizar una presentación de la solución propuesta.

## CAPÍTULO 3: Presentación de la solución propuesta.

### 3.1. Introducción

En este capítulo se comienza la construcción del sistema Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos, generándose todos los artefactos correspondientes siguiendo la metodología de desarrollo de software propuesta: RUP. Se mostrarán los principales conceptos del entorno a través del modelo de dominio así como la descripción textual de cada uno de estos conceptos. También se especifican los requisitos funcionales (RF) y no funcionales (RNF) que el sistema debe cumplir y se representa una muestra de la solución propuesta mediante el diagrama de casos de uso del sistema y las descripciones de los casos de uso.

### 3.2 Modelo de dominio

Debido a que se hace difícil definir procesos de negocio bien estructurados que permitan construir un modelo de negocio completo, se decidió realizar un modelo de dominio.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. **(24)**.

El objetivo fundamental que se persigue al realizar el modelo de dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

#### 3.2.1. Diagrama de clases del Modelo de Dominio

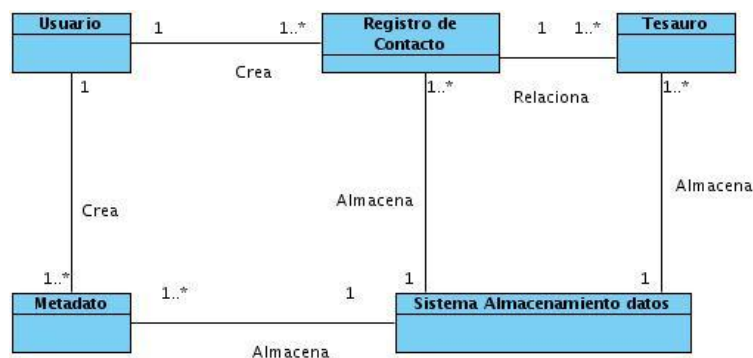


Figura 2: Modelo de Dominio

### 3.2.2. Glosario de Términos del Dominio

**Usuario:** Persona, Organismo u Organización que necesite acceder a los mapas mediante sus metadatos.

**Metadato:** Los metadatos se definen comúnmente como "datos acerca de los datos". Describen el contenido, la calidad, el formato y otras características que llevan asociadas un recurso, constituyendo un mecanismo para caracterizar datos y servicios de forma que usuarios (y aplicaciones) puedan localizarlos y acceder a ellos.

**Registro de Contacto:** El registro de contacto le brinda posibilidad al usuario de registrar todos los datos de este.

**Sistema Almacenamiento Datos:** El sistema de almacenamiento datos es un archivo donde se guardan los datos de la persona que se registró.

**Tesoro:** lenguaje combinatorio que consta de listas de términos que representan un ámbito científico y técnico determinado y que poseen una serie de relaciones semánticas entre los términos que lo conforman.

### 4.3. Modelo del Sistema

Posteriormente se definirán los requisitos funcionales y no funcionales, se describirán los actores y los casos de uso del sistema.

#### 3.3.1. Requisitos Funcionales

Un requisito se puede definir como una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo.

El sistema debe permitir:

**RF1.** Gestionar Registro de Contacto.

*Este requisito brinda la posibilidad al usuario, de registrarse en la aplicación dando paso a realizar las acciones siguientes:*

**RF1.1.** Adicionar Registro de Contacto.

**RF1.2.** Eliminar Registro de Contacto.

**RF1.3.** Modificar Registro de Contacto.

**RF2.** Replicar Registro de Contacto.

*Este requisito brinda la posibilidad al usuario de replicar su registro las veces que desee el mismo*

**RF3.** Importar Registro de Contacto.

*Este requisito brinda la posibilidad al usuario de Importar su registro de contacto en caso de que ya lo haya elaborado en otra ocasión.*

**RF4.** Gestionar datos de Persona.

*Este requisito brinda la posibilidad al usuario después de haberse registrado como contacto, realizar las siguientes acciones:*

**RF4.1.** Adicionar datos de Persona.

**RF4.2.** Eliminar datos de Persona.

**RF5.** Gestionar datos de Organización.

**RF5.1.** Adicionar datos de Organización.

**RF5.2.** Eliminar datos de Organización.

**RF6.** Gestionar datos de Grupo.

**RF6.1.** Adicionar datos de Grupo.

**RF6.2.** Eliminar datos de Grupo.

### **3.3.2. Requisitos no Funcionales**

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

#### **Requisitos de Usabilidad**

**RNF1.** El sistema podrá ser usado por personas con un mínimo conocimiento en el manejo de computadoras.

### **Requisitos de Fiabilidad**

**RNF2.** El sistema debe estar disponible todo el tiempo para sus usuarios, descontando el tiempo en que se encuentre en mantenimiento.

**RNF3.** La herramienta de implementación a utilizar debe tener soporte para recuperación ante fallos y errores.

### **Soporte**

**RNF 4.** La aplicación recibirá mantenimiento en el período de tiempo determinado por el equipo de desarrollo y los clientes.

### **Restricciones de diseño**

**RNF 5.** Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema.

**RNF 6.** Diseño fácil de usar y entendible.

**RNF 7.** El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.

**RNF 8.** Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación).

### **Requisitos para la documentación de usuarios en línea y ayuda del sistema.**

**RNF 9.** El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

### **Requisitos de Interfaz de usuario**

#### **El sistema debe:**

**RNF 10.** Tener una apariencia profesional y un diseño gráfico sencillo, utilizando las buenas prácticas u estándares.

**RNF 11.** Fácil de usar.

### **Requisitos de Hardware**

Para las PCs clientes:

**RNF 12.** Se requiere tengan tarjeta de red.

**RNF 13.** Al menos 64 MB de memoria RAM.

**RNF 14.** Se requiere al menos 100 MB de disco duro.

**RNF 15.** Procesador 512 MHz como mínimo.

**Para los servidores:**

**RNF 16.** Se requiere tarjeta de red.

**RNF 17.** El Servidor de Mapas tenga como mínimo 2GB de RAM y 2GB de disco duro.

**RNF 18.** El Servidor de BD tenga como mínimo 2GB de RAM y 10GB de disco duro.

**RNF 19.** Procesador 3 GHz como mínimo.

### **Requisitos de Software**

La construcción de la aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto, el servidor del usuario final debe tener como requisitos mínimos de software:

**Para las PCs clientes:**

**RNF 20.** Se requiere un Navegador como Mozilla Firefox 3.5.5 u otro navegador que cumpla con los estándares W3C.

**RNF 21.** Se requiere un Sistema operativo: GNU/Linux, Windows o Mac OS.

**Para los Servidores:**

**RNF 22.** Se requiere Sistemas operativos GNU/Linux o Windows Server 2000 o superior.

**RNF 23.** Se requiere un Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsqI incluida.



**RNF 24.** Se requiere PostgreSQL 8.2 como Sistema Gestor de Base de Datos.

### **Requisitos de Licencia**

**RNF 25.** De acuerdo a los tipos de licencias de los componentes y herramientas que se proponen a utilizar para el desarrollo del Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos se, puede catalogar legalmente esta arquitectura de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

### **Requisitos Legales, de Derecho de Autor y otros.**

RNF 26. El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados.

## **3.3.3. Descripción del Sistema Propuesto.**

### **3.3.3.1 Descripción de los actores.**

Un actor no es parte del sistema en desarrollo, es un agente externo que interactúa con el mismo en pos de obtener un resultado esperado. El sistema cuenta con los actores que se especifican a continuación:

<b>Actor</b>	<b>Descripción</b>
<b>Usuario</b>	<b>Es la persona o sistema que utiliza, según el nivel de acceso que tenga, las funcionalidades.</b>

## Diagrama de Casos de Uso del Sistema.

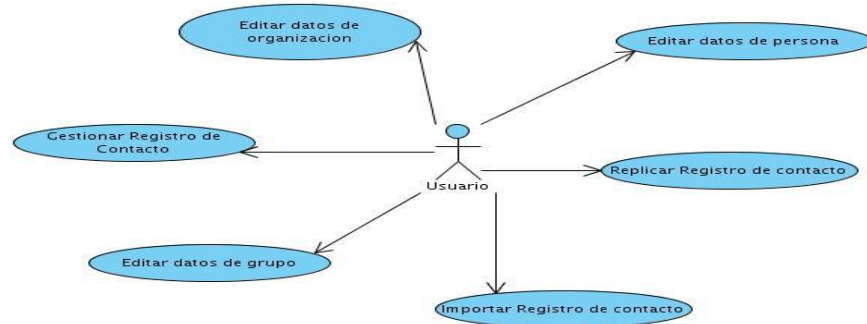


Figura 3: Caso de Uso del Sistema

### 3.3.4 Descripción de los Casos de Uso del Sistema.

#### 3.3.4.1 Descripción Textual del Caso de Uso Gestionar datos de Persona.

Caso de Uso:	Gestionar Registro de Contacto	
Actores:	Usuario	
Propósito	Este caso de uso se lleva a cabo con el objetivo de tener registrado todos los datos de contactos que realizan metadatos.	
Resumen:	El caso de uso se inicia cuando el usuario realiza la acción de registrarse. Este es el CUS que trabaja directamente sobre la agenda, mostrando lo que desee registrar y termina cuando el actor selecciona el campo.	
Precondiciones:	El sistema debe mostrar las opciones referentes al caso de uso.	
Referencias	RF 1,RF 1.1, RF 1.2, RF 1.3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el actor selecciona la opciones de Registrar datos de Contacto :	2. El sistema realiza la operación seleccionada por el actor:	

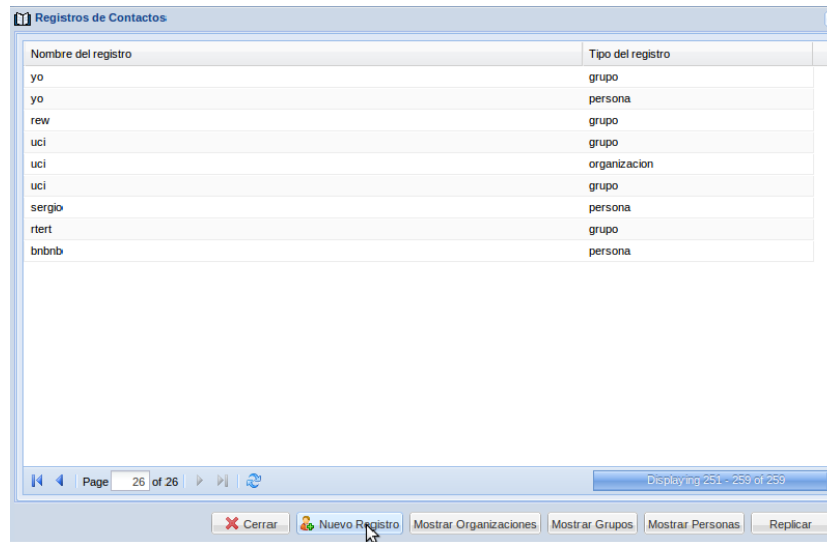
3. El caso de uso termina cuando el sistema procesa la información y da la opción guardar.

Sección "Nuevo registro de contacto"

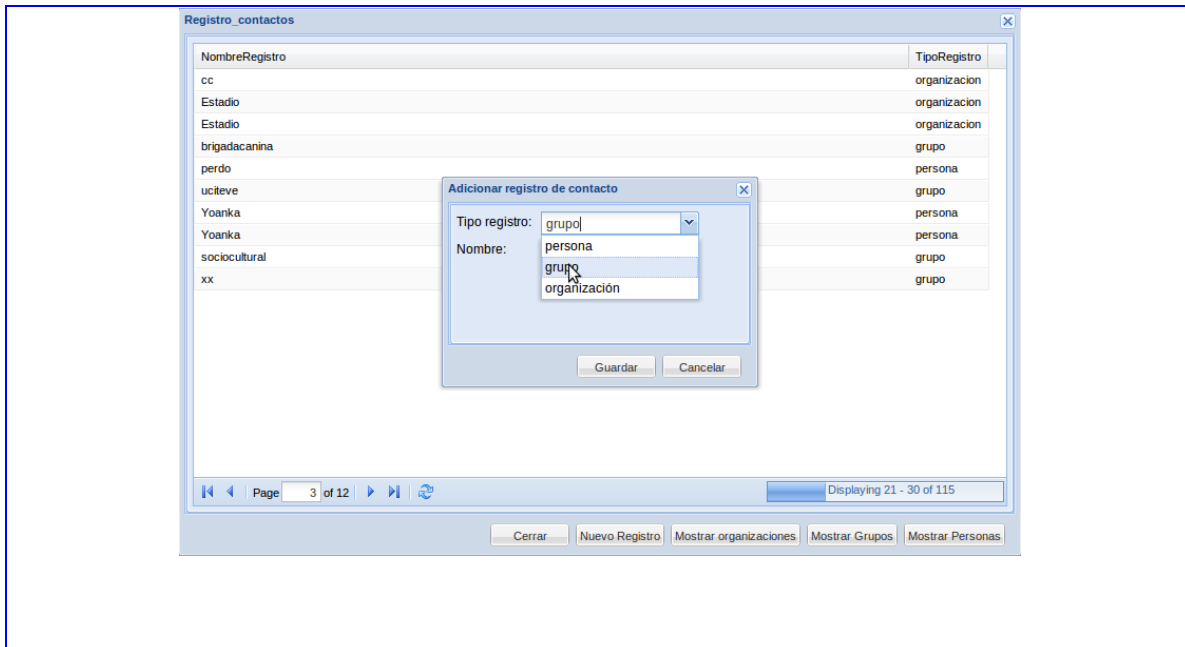
Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción dando clic en el botón Nuevo Registro. (Ver interfaz 1)	2. El sistema muestra el formulario(Ver interfaz 2)
3. El actor llenas los campo del formulario y realiza la opción guardar.	4. El sistema procesa la información y la envía a la base de datos y sierra el formulario

Prototipo de Interfaz

Interfaz 1



Interfaz 2



**Tabla 1: Descripción Textual del Caso de Uso Gestionar datos de Persona.**

Caso de Uso:	Gestionar Datos de Persona
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de tener registrado todos los datos de contactos que realizan metadatos.
Resumen:	El caso de uso se inicia cuando el usuario realiza la acción de editar datos de persona. Este es el CUS que trabaja directamente sobre la agenda, mostrando todos los campos que el actor tiene que llenar y termina cuando el sistema registra los datos.
Precondiciones:	El sistema debe mostrar las opciones referentes al caso de uso.
Referencias	RF 4, RF 4.1 RF 4.2
Prioridad	Crítico
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor una vez registrado el registro de contacto de tipo "persona".	2. El sistema muestra el formulario con los datos a llenar (ver interfaz 3)
	3. El caso de uso termina cuando el sistema procesa la información y da la opción guardar la operación realizada.
<b>Sección "Insertar datos de persona"</b>	
Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción llenando todos los datos del formulario (Ver interfaz 4)	2. El sistema muestra un botón con la opción guardar. (Ver interfaz 5)
3. El actor da clic en la opción Guardar.	4. El sistema procesa la información, la envía a la base de datos
	5. El caso de uso termina cuando el actor da la opción cerrar.
<i>Prototipo de Interfaz</i>	
<i>Interfaz 3</i>	

Adicionar persona

Nombre completo:

Nombre formateado:

Teléfono:

Email:

Grado academico:

Cargo:

Tema de interes:

Pagina web:

Horario atencion:

Instrucciones:

Contacto:

Tipo de recurso:

Fecha:

Guardar Cancelar

Interfaz 4

Adicionar persona

Nombre completo: Sergio Barrera Rodriguez

Nombre formateado: S-B-R

Teléfono: 989325

Email: sbrodriguez@estudiantes.uci.cu

Grado academico: ing.

Cargo: ninguno

Tema de interes: programador

Pagina web: www.seycel.com

Horario atencion: 9

Instrucciones: ninguno

Contacto: por correo

Tipo de recurso: ser

Fecha: 05/25/2011

Guardar Cancelar

Interfaz 5

Guardar Cancelar

Tabla 2: Gestionar Datos de Persona

## ***CAPÍTULO 4: Construcción de la solución propuesta.***

### **4.1. Introducción.**

En este capítulo se describe un ejemplo de los principales artefactos del diseño de la solución propuesta como es el diagrama de clases del diseño del caso de uso Gestionar registro de contacto, el modelo de datos, el modelo de despliegue y el modelo de implementación. Teniendo en cuenta los frameworks definidos para la implementación se explica brevemente su funcionamiento para un mejor entendimiento de la solución, así como los patrones de diseño que se utilizan en la construcción de la misma.

### **4.2. Diseño.**

El modelo de diseño es el modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen un impacto en el sistema a considerar. **(25)**.

También el modelo de diseño sirve de abstracción de la implementación del sistema. Este está muy relacionado con el lenguaje de programación que se emplea por lo que a continuación se realiza una descripción general del funcionamiento de los frameworks utilizados: Symfony y ExtJs, para un mejor entendimiento del flujo descrito en los diagramas de diseño.

#### **4.2.1. Descripción general del funcionamiento del Symfony y ExtJs.**

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC), que está formado por tres niveles: el modelo, la vista y el controlador. Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

En Symfony, la capa del controlador, que contiene el código que liga la lógica de negocio con la presentación, está integrada por los componentes controlador frontal y las acciones. El controlador frontal es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.

Las acciones contienen la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación. El controlador frontal es un

componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática. Las acciones son el corazón de la aplicación, puesto que contienen toda la lógica de la aplicación. Estas utilizan el modelo y definen variables para la vista. Cuando se realiza una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición. Son métodos con el nombre `executeNombreAccion` de una clase llamada `nombreModuloActions` que hereda de la clase `sfActions` y se encuentran agrupadas por módulos. La clase que representa las acciones de un módulo se encuentra en el archivo `actions.class.php`, en el directorio `actions/` del módulo. Las clases de la capa del modelo también se generan automáticamente, en función de la estructura de datos de la aplicación. El componente que se encarga por defecto de gestionar el modelo en Symfony es una capa de tipo ORM (*object/relational mapping*) realizada mediante el proyecto Propel. En las aplicaciones Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realiza mediante objetos; de esta forma nunca se accede de forma explícita a la base de datos. Este comportamiento permite un alto nivel de abstracción y una fácil portabilidad. Por último, la lógica de la vista se puede transformar en un archivo de configuración sencillo, sin necesidad de programarla. La vista se encarga de producir las páginas que se muestran como resultado de las acciones. Esta transforma el modelo en una página Web que le permite al usuario interactuar con ella. Contiene tres partes fundamentales, el layout (es la plantilla global), el complemento de las acciones (llamado plantillas) y las páginas con sus formularios. El layout contiene el código HTML que es común en todas las páginas, para no tener que repetirlo en todas ellas. El contenido de la plantilla se integra en el layout. El complemento de las acciones o plantillas son las encargadas de con los resultados de la acción, insertarse en el cuerpo del layout y generar finalmente la página Web resultado de la petición de un usuario. Las Páginas Clientes y Formularios son las páginas que se generan finalmente y con las que el usuario interactúa. **(22)**.

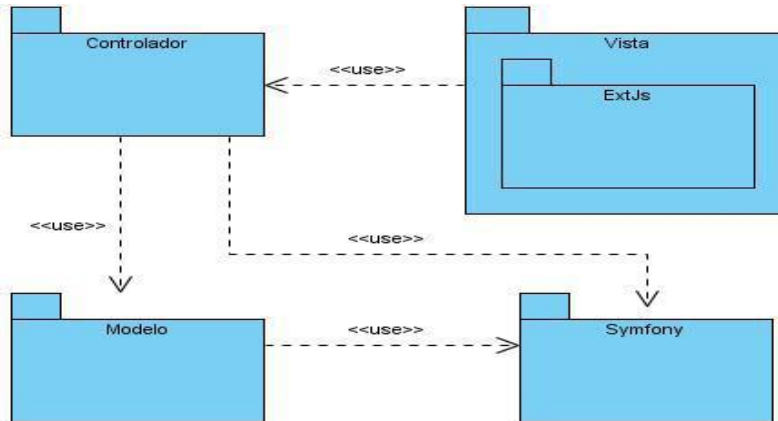
Para el desarrollo de las interfaces del Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos se utiliza EXTJS que es una librería Javascript que incorpora la tecnología AJAX para construir aplicaciones complejas con mayor eficiencia, ello influye en el diseño de la aplicación por tanto es preciso detallar la forma en que interactúa EXTJS y Symfony. La utilización de la librería se expresa a través de un paquete que representa todos los componentes de la propia librería, el paquete `ExtJs`. En cada caso de uso se utilizan un número de ficheros JS comunes a todos, encargados de configurar algunos elementos generales del funcionamiento de la



aplicación. Al mismo tiempo, en cada caso de uso se utilizan otros ficheros JS específicos, los cuales contienen la configuración para las acciones de cada uno de estos en particular.

La estructura general del Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos se puede apreciar a través del siguiente

Diagrama de Paquetes:



**Figura 4: Estructura del Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos.**

### 4.3. Patrones de Diseño.

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. No son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. (24). Symfony implementa un conjunto de patrones de diseño los cuales se describen en este epígrafe, así como los patrones empleados con JavaScript en la interfaz gráfica del sistema.

#### 4.3.1. Patrones GRASP implementados en Symfony.

##### Creador

En la clase Actions se encuentran todas las acciones definidas y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

## **Experto**

Este es uno de los más utilizados, puesto que Doctrine es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

## **Alta Cohesión**

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

## **Controlador**

Todas las peticiones Web son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario. La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

### **4.3.2. Tipos de patrones GOF que implementa Symfony.**

En la categoría Creacionales:

Instancia única (Singleton):

Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a sfContext::getInstance (). En una acción, el método getContext (), es un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony.

Fábrica abstracta (Abstract Factory)

Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea

En la categoría Estructurales:

Objeto compuesto (Composite)

Permite tratar objetos compuestos como si se tratase de uno simple. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera. **(26)**.

#### **4.3.3. Patrones implementados en la vista de la aplicación.**

Fachada (Facade)

La fachada brinda una interfaz sencilla que hace de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. En esta aplicación sirve de intermediario entre la interfaz gráfica y los controladores de Symfony de forma tal que hace el código más legible y por ello puede reducir la dependencia de código externo, ya que la mayoría del código lo usa la fachada, permitiendo así más flexibilidad en el desarrollo del sistema.

#### **4.4. Diagramas de Clases del Diseño.**

Dentro de los artefactos que genera el modelo de diseño se encuentra el Diagrama de Clases del Diseño, en el cual se muestran un conjunto de clases, interfaces y colaboraciones, así como las relaciones entre sí. Se realiza cada diagrama por caso de uso identificado, estos diagramas son importantes pues visualizan, especifican y documentan modelos estructurales, lográndose una muestra amplia y confiable del sistema previo a su implementación. En este epígrafe se muestra el diagrama de clases del diseño del Caso de Uso Gestionar registro de Contacto y el paquete Modelo correspondiente a este.

#### 4.4.1. Realización del Caso de Uso: Gestionar Registro de Contacto

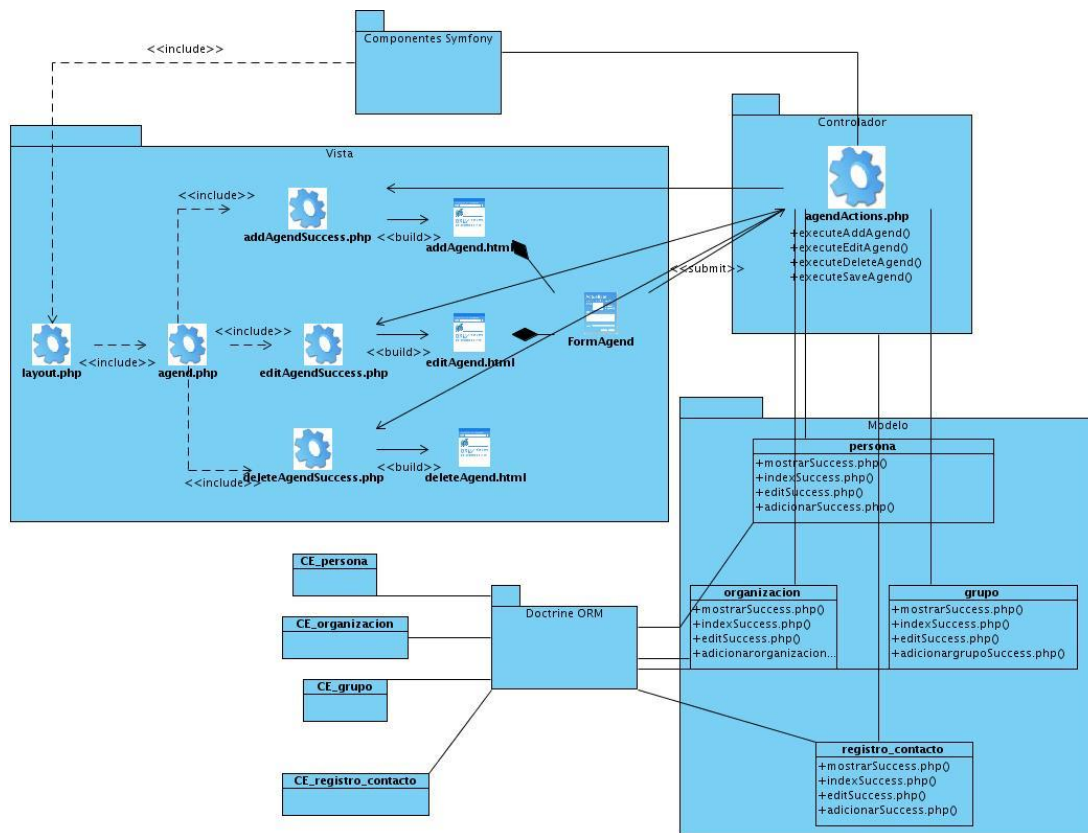


Figura 5: Diagrama de Clases del Diseño CU Gestionar Registro de Contacto

#### 4.5. Diseño de la Base de Datos.

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Para modelar el diseño de la base de datos primeramente se realiza el diagrama de clases persistentes a partir de las clases del diseño que tienen un carácter permanente y luego se lleva a cabo la realización del modelo de datos del sistema. Un Modelo de Datos se puede definir como un conjunto de conceptos, reglas y convenciones bien definidos que permiten aplicar una serie de abstracciones a fin de describir y manipular los datos de un cierto mundo real que se desea almacenar en la base de datos. A continuación se presenta el diagrama de clases persistentes y el modelo entidad-relación correspondiente a las entidades persona, grupo u organización para una mejor visualización y comprensión de los mismos.

#### 4.5.1. Diagrama de Clases Persistentes.

Vista Detallada: Entidad Persona.

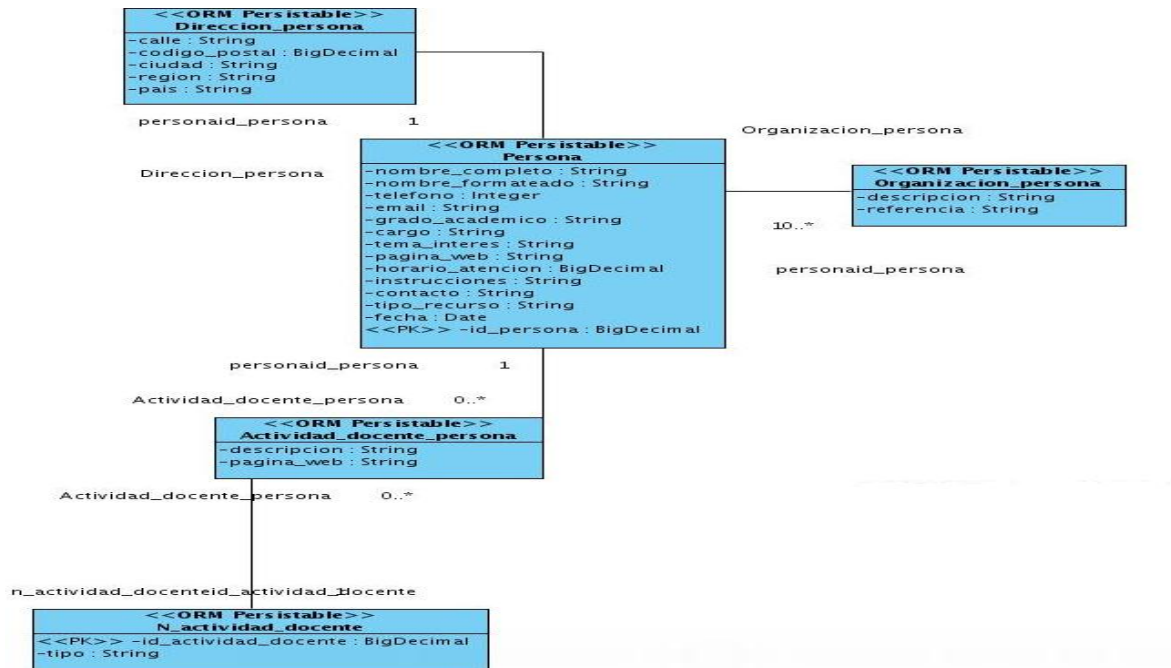


Figura 6: Diagrama de Clases Persistentes. Paquete Información del Módulo para la Agenda de Contactos

Vista Detallada: Entidad Grupo.

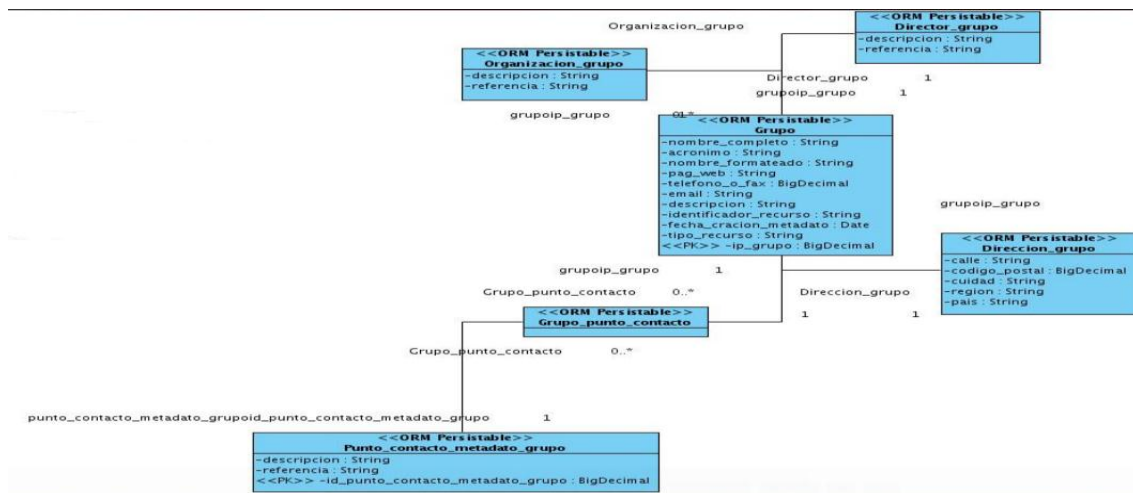


Figura 7: Diagrama de Clases Persistentes. Paquete Información del Módulo para la Agenda de Contactos

### Vista Detallada: Entidad Organización.

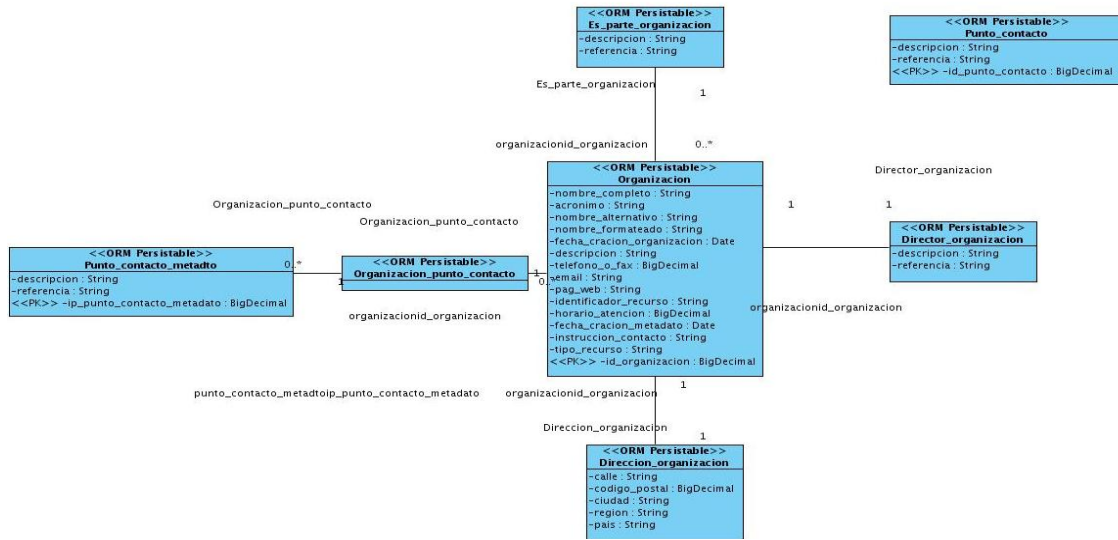
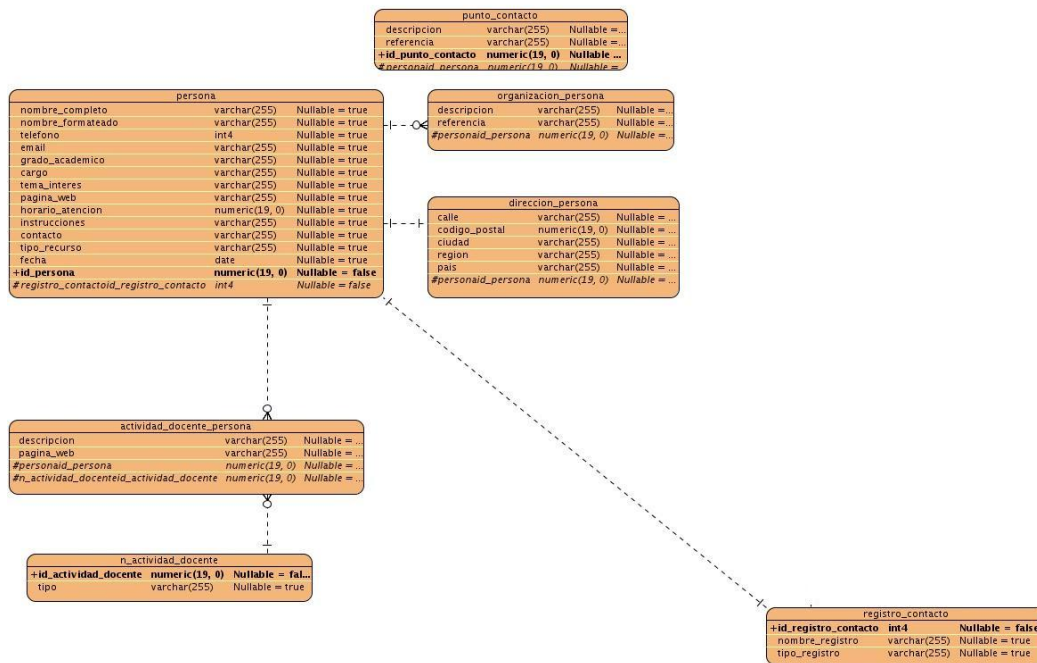


Figura 8: Diagrama de Clases Persistentes. Paquete Información del Módulo para la Agenda de Contactos

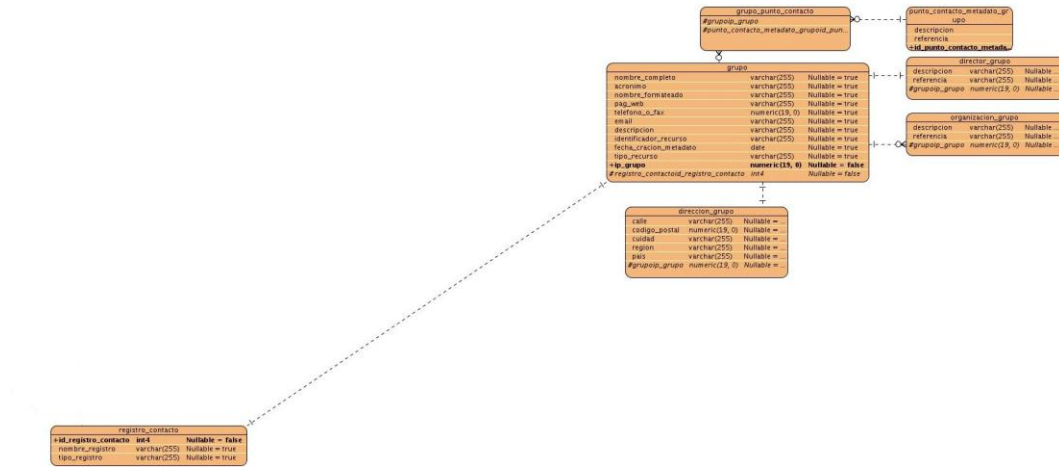
### 4.5.2. Modelo Entidad-Relación.

### Vista Detallada: Entidad Persona.



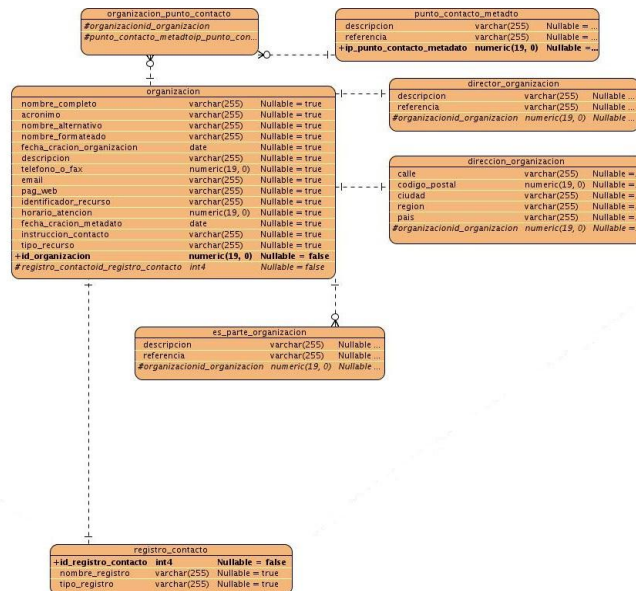
**Figura 9: Modelo Entidad Relación. Paquete Información del Módulo para la Agenda de Contactos**

Vista Detallada: Entidad Grupo.



**Figura 10: Modelo Entidad Relación. Paquete Información del Módulo para la Agenda de Contactos**

Vista Detallada: Entidad Organización.



**Figura 11: Modelo Entidad Relación. Paquete Información del Módulo para la Agenda de Contactos**

#### 4.6. Modelo de Despliegue.

El Modelo de Despliegue es un modelo de objetos que representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. **(23)**.

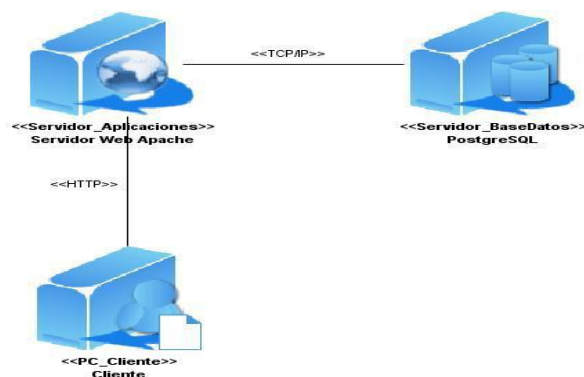


Figura 12: Diagrama de Despliegue.

#### 4.7. Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros. **(24)** A continuación se muestra una vista general del modelo de implementación del Módulo para la Gestión de la Agenda de Contactos del Sistema para la Gestión de Metadatos Geográficos.



Vista General.

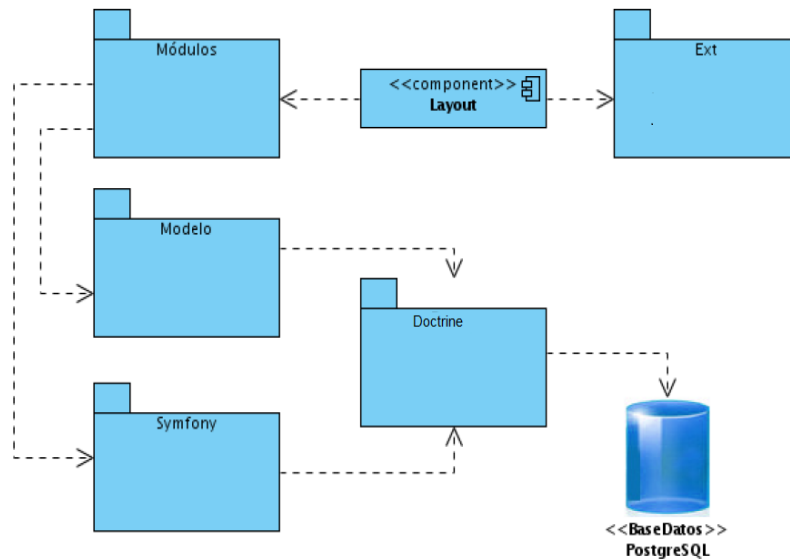


Figura 13: Diagrama de Componentes. Vista General.

Para una mayor comprensión del Diagrama de Componentes, se expone una vista detallada de los paquetes Modelo, Ext, Módulos y los componentes que contiene cada Módulo en particular. *Vista Detallada: Paquete Modelo.*

*Vista Detallada: Paquete Modelo.*

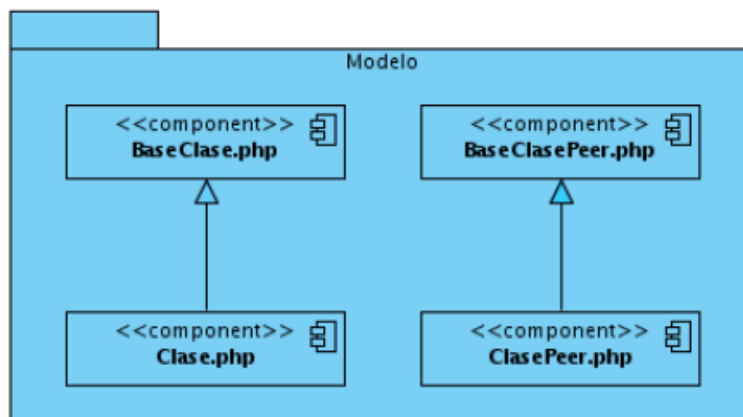


Figura 14: Diagrama de Componentes. Paquete Modelo.

Vista Detallada: Paquete Ext.

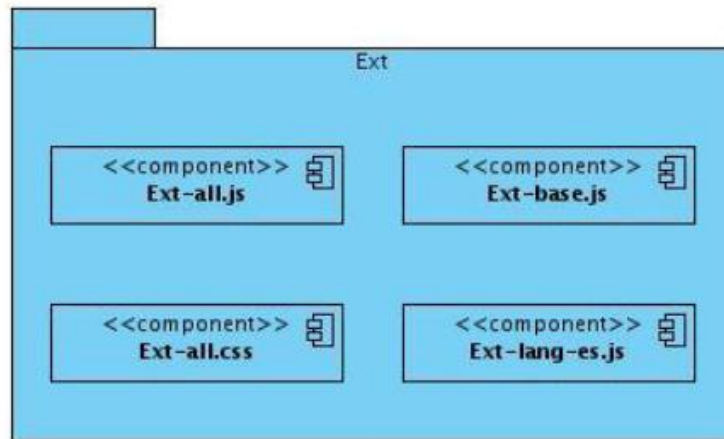


Figura 15: Diagrama de Componentes. Paquete Ext.

Vista Detallada: Paquete Módulos

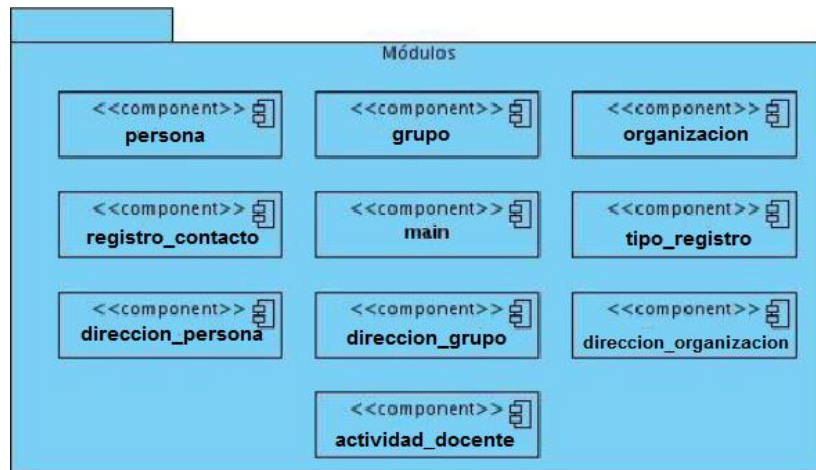


Figura 16: Diagrama de Componentes. Paquete Módulos.

### Vista Detallada de cada Módulo

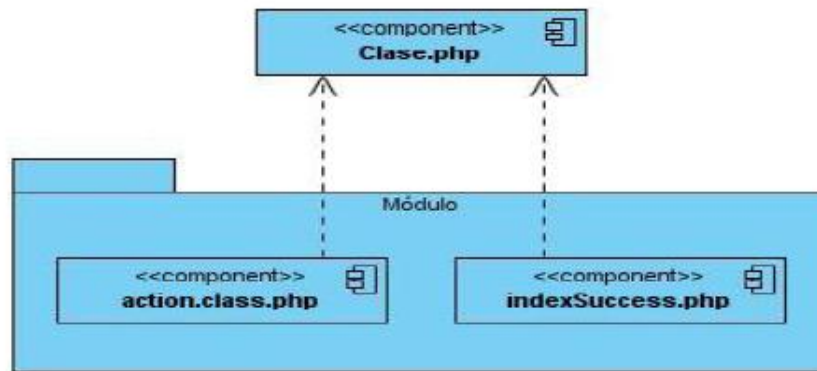


Figura 17: Diagrama de Componentes. Paquete Módulo.

#### 4.8. Conclusiones.

Durante el desarrollo de este capítulo se llevaron a cabo los flujos de trabajo: Análisis y Diseño e Implementación. Como resultado de estos flujos se generaron los artefactos pertinentes a cada uno de ellos como son el Modelo de Diseño, el cual permitió representar la estructura del Módulo para la Agenda de Contactos para la gestión de metadatos geográficos mostrando sus clases, atributos y las relaciones entre ellos. Se realizó el Modelo de Despliegue a través del cual se muestra la distribución física del sistema y sus conexiones. Mediante la realización del Modelo de Datos se logró describir los elementos que intervienen en el problema dado y la forma en que se relacionan estos elementos entre sí. Finalmente se expuso el Modelo de Implementación, donde se especificaron los distintos componentes creados para desarrollar la aplicación y la relación entre ellos, lográndose traducir el diseño en términos de componentes ejecutables. De esta forma se cuenta con la base principal en la cual apoyar la construcción del sistema.

#### 4.9. Validación de la solución propuesta.

Las pruebas de un software están constituidas por un conjunto de herramientas, técnicas y métodos que tributan a la excelencia del desempeño de un programa. Técnicamente consisten en inspeccionar manualmente el código o hacer pruebas de ensayos (ejecutar el software y ver los resultados) y el objetivo principal es encontrar errores o defectos en el software.

A continuación se muestra el diseño de casos de prueba de caja negra de los Casos de Usos Gestionar registro de Contacto y Gestionar datos de Persona. Los diseños de caso de pruebas correspondientes a los demás Casos de Uso pueden ser consultados en el Anexo. Para ello solo es necesario conocer la interfaz y tratar de probar cada uno de los elementos que componen a la misma.

##### 4.9.1. Caso de prueba del Casos de uso Gestionar registro de Contacto.

###### Descripción general

El caso de uso inicia cuando el usuario selecciona la opción “Nuevo Registro”, esto lo hace dando clic en el botón “Nuevo Registro” de la interfaz general. Seguidamente se abre una nueva interfaz que es la que le dará paso a ejecutar la acción deseada después de haber llenado los campos, y termina cuando realiza la acción guardar.

###### Sección a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
CS.1 Gestionar registro de Contacto	EC 1.1: Interfaz principal llamada “Registros de contactos”	El sistema brinda la posibilidad al usuario de gestionar su registro de contacto

	EC 1.2 Tratamiento de errores cuando se gestiona un registro de contacto	El sistema brinda la posibilidad al usuario de avisarle cuando un campo es llenado incorrectamente
--	--	--

**Tabla 3: Sección a probar en el Caso de Uso**

**Descripción de Variables**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Tipo Registro	Selección de la opción (ComboBox)	No	Se debe seleccionar una de las opciones siguientes: Persona Grupo
[2]	Nombre	Campo de texto (TextBox)	No	Se escribe el nombre del registro seleccionado

**Tabla 4: Descripción de Variables**

**Matriz de Datos**

Escenario	Variable de Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central

Gestionar registro de Contacto	T	El sistema brinda la posibilidad al usuario de gestionar su registro de contacto	Satisfactorio	<p>Paso 1: Se brinda la posibilidad de gestionar un registro de contacto</p> <p>Paso 2: El sistema muestra la interfaz donde se gestionara el registro</p> <p>Paso 3: El usuario llena los campos pedidos y realiza la opción guardar</p>
Tratamiento de errores cuando se gestiona un registro de contacto	N	El sistema brinda la posibilidad de avisarle al usuario cuando ha entrado algún campo incorrecto	Satisfactorio	<p>Paso 1: El usuario llena el formulario.</p> <p>Paso 2: El usuario llena el formulario con datos incorrectos.</p> <p>Paso 3: El sistema señala el dato que el usuario lleno incorrectamente y lo marca en rojo</p>

**Tabla 5: Matriz de Datos**

#### **4.9.2. Caso de prueba del Casos de uso Gestionar datos de Persona.**

##### **Descripción general**

El caso de uso inicia cuando el usuario ya termino de gestionar su registro de contacto y que sea del tipo “persona”. Seguidamente se abre una nueva interfaz donde aparece el formulario donde el usuario debe de llenar todos sus datos y el caso de uso termina cuando realiza la opción guardar dando clic en el botón con el mismo nombre.

##### **Sección a probar en el Caso de Uso**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
----------------------	--------------------------	---------------------------------

CS.1 Gestionar datos de persona	EC 1.1: Formulario	El sistema brinda la posibilidad al usuario de llenar sus datos y guárdalos
	EC 1.2 Tratamiento de errores cuando se gestiona datos de persona	El sistema brinda la posibilidad al usuario de avisarle cuando un campo es llenado incorrectamente

### Descripción de Variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Formulario	Campos de textos (TextBox)	No	Se debe de llenar todos los datos del formulario correctamente

### Matriz de Datos

Escenario	Variable de Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Gestionar datos de Persona	T	El sistema brinda la posibilidad al usuario de gestionar su datos	Satisfactorio	<p>Paso 1: El sistema muestra el formulario con todos los campo a llenar</p> <p>Paso 3: El usuario llena los campos pedidos y realiza la opción guardar</p>

<p>Tratamiento de errores cuando se gestiona datos de persona</p>	<p>N</p>	<p>El sistema brinda la posibilidad de avisarle al usuario cuando ha entrado algún campo incorrecto</p>	<p>Satisfactorio</p>	<p>Paso 1: El usuario llena el formulario.  Paso 2: El usuario llena el formulario con datos incorrectos.  Paso 3: El sistema señala el dato que el usuario lleno incorrectamente y lo marca en rojo</p>
---	----------	---	----------------------	--



## ***CONCLUSIONES***

Con la proliferación de las redes sociales se han establecido variadas plataformas que fomentan la colaboración y la aplicación de herramientas que permiten crear canales de comunicación e interacción entre las personas con algún tipo de interés o necesidad específica. La agenda de contactos es un ejemplo más de las herramientas creadas para registrar información de personas u organizaciones que posibilitan tener almacenado datos de interés. Durante el desarrollo de esta investigación se logró cumplir con el objetivo general propuesto: “Desarrollar un módulo para el Sistema para la Gestión de Metadatos Geográficos que permita la gestión de la información de sus contactos”. El mismo, permitirá acceder a la misma de manera más eficiente y rápida.

El Sistema para la Gestión de Metadatos Geográficos tendrá una manera sencilla y eficiente de registrar a toda aquella persona u organización que realice algún metadato geográfico, esto posibilita una mayor calidad del producto existente ya que el usuario tendrá la forma de no perder la información de sus contactos y tenerla registrada en caso de necesidad. El modulo está desarrollado con una arquitectura web que posibilita explotar los beneficios de la red, minizando los costos de hardware en las organizaciones cubanas. No obstante, el alcance del producto no debe limitarse al territorio nacional, ya que es potencialmente exportable tanto como solución independiente o integrada a un SIG como lo es GeneSIG. La aplicación fue desarrollada mediante herramientas multiplataforma lo que permite que pueda ser desplegada sobre entornos libres, de esta forma se garantiza un producto con mayor tiempo de vida, mas reutilizable y eficiente. Además, le permitirá al país contar con una herramienta propia que responda a los intereses de sus organizaciones. Teniendo en cuenta estos aspectos, y además que Cuba está inmersa en la creación de una infraestructura de Datos espaciales, contar con una herramienta para creación de metadatos propias y además con la posibilidad de almacenar todos los datos de sus registros de contacto es un gran reto y oportunidad, que sin duda posee un impacto social considerable, ya que no existe otra solución cubana similar. Esta además ayuda en la informatización de las esferas socioeconómicas del país, que hoy se sustentan en software propietario para desarrollar estas tareas, y en peores casos para suplir el trabajo manual de algunas entidades que presentan un deterioro en su productividad y desempeño, por falta de sistemas que automaticen sus labores y apoyen la toma de decisiones, muchas veces de gran impacto en la sociedad.

## ***RECOMENDACIONES***

- Se recomienda ejercerle pruebas de Caja Blanca a la aplicación para una mejor seguridad de la misma.

## **REFERENCIA BIBLIOGRÁFICA**

(1) *Real Academia Española*. (s.f.). Recuperado el 20 de noviembre de 2010, de <http://buscon.rae.es/drael>

(2) *sedic*. (s.f.). Recuperado el 29 de noviembre de 2010, de <http://www.sedic.es/autoformacion/metadatos/tema1.htm>

(3) COMISIÓN DE LAS COMUNIDADES EUROPEAS. (2008). *Reglamento*.

(4) Cisneros Díaz, L. R. (1194). LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG: DEFINICIÓN, CARACTERÍSTICAS, ESTADO ACTUAL Y TENDENCIAS DE DESARROLLO. *Revista Internacional de Ciencias de la Tierra*.

(5) Gómez, J. (s.f.). *softonic*. Recuperado el 29 de noviembre de 2010, de “Agenda de contactos gratuita con funciones de CRM”: <http://open-contacts.softonic.com/>

(6) Vani.be. (s.f.). *contactkeeper*. Recuperado el 29 de noviembre de 2010, de <http://www.vani.be/contactkeeper/>

(7) *hormiga.org*. (s.f.). Recuperado el 29 de noviembre de 2010, de <http://www.hormiga.org/programas-gratis/Agenda-XP-2-5-2.htm>

(8) *Planeta Freeware*. (s.f.). Recuperado el 29 de noviembre de 2010, de <http://planetafreeware.blogspot.com/2010/09/myportablepim-130.html>

(9) *programas.com*. (s.f.). Recuperado el 29 de noviembre de 2010, de <http://agenda-msd-multiusuario.programas.com/>

(10) Madrid, D. B. *Prácticas con CatMDEdit*. España.

(11) CHACÓN, JULIO CÉSAR RUEDA. 2006. APLICACIÓN DE LA METODOLOGÍA RUP PARA EL DESARROLLO RÁPIDO DE APLICACIONES BASADO EN EL. Guatemala: Universidad de San Carlos de Guatemala. Facultad de Ingeniería. Escuela de Ingeniería en Ciencias, 2006.

(12) *METODOLOGIA de desarrollo de SOfware*. Cuba : s.n., 15/10/2008.

(13) Orallo, Enrique Hernández. 2007. El Lenguaje Unificado de Modelado (UML). 2007.

(14) El Lenguaje Unificado de Modelado. 2009.

(15) Fernández, Sagrario Peralta. 2008. El servidor Web. Arquitectura y funcionamiento. [En línea] 2008. [Citado el: 3 de diciembre de 2009.]

<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=366>.

(16) Apache. 2007. Servidor web Apache. [En línea] 2007. [Citado el: 3 de diciembre de 2009.]

<http://www.apache.org>.

(17) PHP. [En línea] [Citado el: 10 de 2 de 2011.] [peru.blogalaxia.com/post/hypertext](http://peru.blogalaxia.com/post/hypertext).

(18) Rodas, Raúl Hinostraza. 2007. [www.linuxcentro.net](http://www.linuxcentro.net). Usuarios y Grupos en Linux. [En línea] 2007. [Citado el: 21 de diciembre de 2009.]

(19) Sencha Solutions. 2008. ExtJs. Documentación Ext JS JavaScript Library. [En línea] 2008. [Citado el: 21 de diciembre de 2009.] <http://extjs.com/deploy/dev/docs>.

(20) Eguíluz, Javier Pérez. 2006. LibrosWeb.es. Introducción a AJAX. [En línea] 2006. [Citado el: 22 de diciembre de 2009.] <http://librosweb.es/ajax/index.html>.

(21) Visual Paradigm International. 2010. Visual Paradigm. Visual Paradigm for UML - UML tool for software application development. [En línea] 11 de Enero de 2010. [Citado el: 19 de diciembre de 2009.] <http://www.visual-paradigm.com/product/vpuml/>.

Wheeler, Veronika. 2008. Linux en Español. Entorno de desarrollo integrado. [En línea] 26 de marzo de 2008. [Citado el: 20 de diciembre de 2009.] <http://www.linuxespanol.com/viewtopic.php?p=96480&sid=c1e2f61c2c9f25c5ca0e492a6c87c62e>.

(22) Potencier, Fabien y Zaninotto, François. 2008. Symfony la guía definitiva. 2008.

(23) Wheeler, Veronika. 2008. Linux en Español. Entorno de desarrollo integrado. [En línea] 26 de marzo de 2008. [Citado el: 20 de diciembre de 2009.]

<http://www.linuxespanol.com/viewtopic.php?p=96480&sid=c1e2f61c2c9f25c5ca0e492a6c87c62e>

**(24)** Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. Madrid: Series Editors, 2000.

**(25)** Torres Gastelú, Dr. Carlos A. 2007. Universidad Veracruzana Facultad de Administración. Sistemas Computacionales Administrativos E.E Base de Datos. Apache e Internet Information Services. [En línea] 2007. [Citado el: 20 de diciembre de 2009.] <http://www.slideshare.net/equipo9/apache-e-iis>.

**(26)** Rabaix, Thomas. 2008. Patrones de diseño que implementa Symfony. [En línea] 2008. [Citado el: 15 de Enero de 2010.] <http://www.mail-archive.com/symfony-users@googlegroups.com/msg07161.html>.

**(27)** Popkin Software and Systems. 2008. Modelado de Sistemas con UML. 2008.

## **BIBLIOGRAFIA**

1. CE. 2008. COMISIÓN DE LAS COMUNIDADES EUROPEAS. REGLAMENTO (CE) No 1205/2008 DE LA COMISIÓN de 3 de diciembre de 2008 por el que se ejecuta la Directiva. 2008.
2. DICCIONARIO DE LA LENGUA ESPAÑOLA. Vigésima segunda edición. [En línea] [Citado el: 17 de noviembre de 2010.] <http://buscon.rae.es/drae/>.
3. Franco, Rodolfo. 2009. Universidad Distrital Francisco Jose de Caldas. [En línea] 11 de agosto de 2009. [Citado el: 17 de noviembre de 2010.] <http://www.udistrital.edu.co/comunidad/profesores/rfranco/metadatos.htm>.
4. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. Madrid : Series Editors, 2000.
5. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA SIG: DEFINICIÓN, CARACTERÍSTICAS, ESTADO ACTUAL Y TENDENCIAS DE DESARROLLO. Cisneros, Luís R. Díaz y Duffatt, Rafael Candeaux. Julio de 1994. s.l. : Mapping Interactivo. Revista Internacional de Ciencias de la Tierra., Julio de 1994, Vols. ISSN: 1.131-9.100, s.l. Bibliografía **69**
6. Potencier, Fabien y Zaninotto, François. 2008. Symfony la guía definitiva. 2008.
7. Programa Geodato. 2007. Procedimiento para la Documentacion del Metadato. s.l. s.l. : Geominera Oriente, 2007.
8. Sencha Solutions. 2008. ExtJs. Documentación Ext JS JavaScript Library. [En línea] 2008. [Citado el: 7 de enero del 2011.] <http://extjs.com/deploy/dev/docs>.
9. Thompson, Ivan. 2008. Portal de Mercadotecnia. [En línea] Octubre de 2008. [Citado el: 15 de Noviembre de 2010.] <http://www.promonegocios.net/mercadotecnia/definicion-informacion.html>.
10. Zayas, Dr. Cs. Carlos Alvarez de. 1995. METODOLOGIA DE LA INVESTIGACION CIENTIFICA. Santiago de Cuba : CENTRO DE ESTUDIOS DE EDUCACION SUPERIOR "MANUEL F. GRAN", 1995.

# ANEXOS

## Anexo 1

Caso de Uso:	Gestionar Datos de Grupo
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de tener registrado todos los datos de contactos que realizan metadatos.
Resumen:	El caso de uso se inicia cuando el usuario realiza la acción de registrarse. Este es el CUS que trabaja directamente sobre la agenda, mostrando lo que desee registrar y termina cuando el actor selecciona el campo.
Precondiciones:	El sistema debe mostrar las opciones referentes al caso de uso.
Referencias	RF 1,RF 1.1, RF 1.2, RF 1.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor una vez registrado el registro de contacto de tipo "grupo".	2. El sistema muestra el formulario donde se encuentran los datos a llenar
	3. El caso de uso termina cuando el sistema procesa la información y da la opción aceptar.
Sección "Adicionar datos de Grupo"	
Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción llenando todos los datos del formulario (ver interfaz 1)	2. El sistema muestra la opción de guardar los datos (ver interfaz 2)

3. El actor selecciona la opción guardar.

4. El sistema procesa la información y la envía para la base de datos

### Prototipo de Interfaz

#### Interfaz 1

A screenshot of a web form titled "Adicionar grupo". The form contains the following fields and values:

- Nombre completo: Metadatos Geograficos
- Acronimo: M-G
- Nombre formateado: MetaGeop
- Pagina web: http://www.meta.com.cu
- Teléfono ó Fax: 6555655
- Email: meta@geografic.cu
- Descripcion: Especialistas
- Identificador del recurso: M
- Fecha creación metadato: 06/21/2011
- Tipo de recurso: maximo|

At the bottom of the form, there are three buttons: "Buscar...", "Guardar" (with a floppy disk icon), and "Cancelar" (with a red X icon). A mouse cursor is pointing at the "Guardar" button.

#### Interfaz 2

A screenshot of a web form showing only the bottom buttons: "Buscar...", "Guardar" (with a floppy disk icon), and "Cancelar" (with a red X icon). A mouse cursor is pointing at the "Guardar" button.

### Sección "Eliminar datos de Grupo "

Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción dando clic encima del registro que desee borrar. (Ver Interfaz 3).	2. El sistema muestra la opción de Eliminar Registro en un botón. (Ver interfaz 4).
3. El actor selecciona la opción Eliminar Registro	4. El sistema muestra una ventana para que el actor decida si desea borrar o no el registro.



5. El actor selecciona Sí o NO.	6. Si el actor selecciona la opción Si el sistema borra el registro y si selecciona la opción No el sistema no realiza ningún cambio.
	5. El caso de uso termina cuando el sistema realiza la operación seleccionada

*Prototipo de Interfaz*

*Interfaz 3*

Nombre Completo	Acronimo	Nombre Formateado	Pagina Web	Telefono/Fax
Metadatos Geograficos	M-G	MetaGeop	http://www.meta.i	6555655
titititi	tt	t	tt	67676

*Interfaz 4*

Nombre Completo	Acronimo	Nombre Formateado	Pagina Web	Telefono/Fax
Metadatos Geograficos	M-G	MetaGeop	http://www.meta.i	6555655
titititi	tt	t	tt	67676
asdsdad	adasd	asdasd	asdsd	4353
ll	ll	llllll	wewew	34343
Brigada Canina	brica	B-C	skjkdjfkdsj.kjsjds	676767
UCI-TEVE	UCITEVE	U-CT-V	www.uciteve.com	3435
paployeduanys	pe	pp-ee	opopo	2323
alogandre	alo	a-lo	alogrande.com	3434
	g	gg	99999	
	gg	99999	99999999	

Page 1 of 3 | Displaying 1 - 10 of 27

Cerrar Editar Eliminar

Caso de Uso:	Gestionar Datos de Organización
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de tener registrado todos los datos

	de contactos que realizan metadatos.
Resumen:	El caso de uso se inicia cuando el usuario realiza la acción de registrarse. Este es el CUS que trabaja directamente sobre la agenda, mostrando lo que desee registrar y termina cuando el actor selecciona el campo.
Precondiciones:	El sistema debe mostrar las opciones referentes al caso de uso.
Referencias	RF 1,RF 1.1, RF 1.2, RF 1.3
Prioridad	Crítico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor guarda el registro de contacto de tipo "Organización".	2. El sistema muestra el formulario donde se encuentran los datos a llenar
	3. El caso de uso termina cuando el sistema procesa la información y da la opción aceptar.

#### Sección "Adicionar datos de Organización "

Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción llenando todos los datos del formulario (ver interfaz 1)	2. El sistema muestra la opción de guardar los datos (ver interfaz 2)
6. El actor selecciona la opción guardar.	4. El sistema procesa la información y la envía para la base de datos

#### Prototipo de Interfaz

*Interfaz 1*

Adicionar Organizacion

Nombre completo: Organizacion Mundial Geografic Metadata

Acronimo: O-M-G-M

Nombre alternativo: Mundial Geografic

Nombre formateado: geometa

Fecha creación organizacion: 06/15/2002

Descripcion: Especializada

Tel&efono o Fax: 4457642

Email: geometa@tvvf.com

Pagina web: http://geometa.com

Identificador del recurso: geometa

Horario de atencion: 18

Fecha creación metadato: 06/16/2010

Instruccion contacto: por telefono

Tipo de recurso: meta

Buscar... Guardar Cancelar

Interfaz 2

Buscar... Guardar Cancelar

Sección "Eliminar datos de Grupo "

Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción dando clic encima del registro que desee borrar. (Ver Interfaz 3).	2. El sistema muestra la opción de Eliminar Registro en un botón. (Ver interfaz 4).
3. El actor selecciona la opción Eliminar Registro	4. El sistema muestra una ventana para que el actor decida si desea borrar o no el registro .(Ver Interfaz 5)
5. El actor selecciona Sí o NO.	6. Si el actor selecciona la opción Si el sistema borra el registro y si selecciona la opción No el sistema no realiza ningún cambio.
	8. El caso de uso termina cuando el sistema realiza la operación

seleccionada

### Prototipo de Interfaz

#### Interfaz 3

Nombre Completo	Acronimo	Nombre Alternativo	Nombre Formateado	Fecha Cracion de la Organizacion
Organizacion Mundial Geografic Metadata	O-M-G-M	Mundial Geografic	geometa	2002-06-15

#### Interfaz 4

Nombre Completo	Acronimo	Nombre Alternativo	Nombre Formateado	Fecha Cracion de la Organizacion
Organizacion Mundial Geografic Metadata	O-M-G-M	Mundial Geografic	geometa	2002-06-15
oooooooooooooooooooo	oooo	oooooo	oooo	2011-05-02
rerererere	rer	re	rerr	2011-05-01
	est.capsan	tsuami	C-S	2011-05-10
ccccc	cc	c	c	2011-05-03
	v	v	v	2011-05-04
	z	z	z	2011-05-11
	z	z	z	2011-05-11
	h	h	h	2011-05-18
	h	h	h	2011-05-18

Caso de Uso:	Importar Registro de Contacto
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de tener registrado todos los datos de contactos que realizan metadatos.
Resumen:	El caso de uso se inicia cuando el usuario realiza la acción de registrarse. Este es el CUS que trabaja directamente sobre la agenda, mostrando lo que desee registrar y termina cuando el actor selecciona el campo.
Precondiciones:	El sistema debe mostrar las opciones referentes al caso de uso.
Referencias	RF 1,RF 1.1, RF 1.2, RF 1.3

Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor una vez registrado el registro de contacto ya sea de tipo "grupo", "organización", o "persona".	2. El sistema muestra el formulario donde se encuentran los datos a importar
	3. El caso de uso termina cuando el sistema procesa la información y da la opción aceptar.
Sección "Importar Registro de contacto"	
Acción del Actor	Respuesta del Sistema
1. El actor inicia la acción dando clic en el botón buscar	2. El sistema muestra la opción de buscar la dirección del fichero.
3. El actor selecciona la opción aceptar.	4. El sistema procesa la información y la envía para la base de datos y llena los campos del formulario

Caso de Uso:	Replicar Registro de Contacto
Actores:	Usuario
Propósito	Este caso de uso se lleva a cabo con el objetivo de que el usuario pueda replicar su registro de contacto.
Resumen:	El caso de uso se inicia cuando el usuario realiza la acción de registrarse. Este es el CUS que trabaja directamente sobre la agenda, mostrando lo que desee registrar y termina cuando el actor selecciona el campo.
Precondiciones:	El sistema debe mostrar las opciones referentes al caso de uso.

Referencias	RF 1,RF 1.1, RF 1.2, RF 1.3
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor una vez registrado el registro de contacto ya sea de tipo "grupo", "organización", o "persona".	2. El sistema muestra la opción de replicar registro en un botón.
	3. El caso de uso termina cuando el sistema procesa la información y da la opción aceptar.
Sección "Replicar Registro de contacto"	
Acción del Actor	Respuesta del Sistema
1. El actor inicia seleccionando el registro con clic sobre el.	2. El sistema muestra la opción replicar registro
4. El actor selecciona la opción mostrada.	4. El sistema procesa la información, la envía para la base de datos muestra el registro replicado

## ANEXOS

### Anexo 2

#### Caso de prueba del Casos de uso Gestionar datos de Grupo.

#### Descripción general

El caso de uso inicia cuando el usuario ya termino de gestionar su registro de contacto y que sea del tipo "grupo". Seguidamente se abre una nueva interfaz donde aparece el formulario donde el usuario debe de llenar todos sus datos y el caso de uso termina cuando realiza la opción guardar dando clic en el botón con el mismo nombre.

#### Sección a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
CS.1 Gestionar datos de grupo	EC 1.1: Formulario	El sistema brinda la posibilidad al usuario de llenar sus datos y guárdalos
	EC 1.2 Tratamiento de errores cuando se gestiona datos de grupo	El sistema brinda la posibilidad al usuario de avisarle cuando un campo es llenado incorrectamente

#### Descripción de Variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Formulario	Campos de textos (TextBox)	No	Se debe de llenar todos los datos del formulario correctamente

#### Matriz de Datos

Escenario	Variable de Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Gestionar datos de Grupo	T	El sistema brinda la posibilidad al usuario de gestionar su datos	Satisfactorio	Paso 1: El sistema muestra el formulario con todos los campo a llenar  Paso 3: El usuario llena los campos pedidos y realiza la opción guardar
Tratamiento de errores cuando se gestiona datos de grupo	N	El sistema brinda la posibilidad de avisarle al usuario cuando ha entrado algún campo incorrecto	Satisfactorio	Paso 1: El usuario llena el formulario.  Paso 2: El usuario llena el formulario con datos incorrectos.  Paso 3: El sistema señala el dato que el usuario lleno incorrectamente y lo marca en rojo

### **Caso de prueba del Casos de uso Gestionar datos de Organización.**

#### **Descripción general**

El caso de uso inicia cuando el usuario ya termino de gestionar su registro de contacto y que sea del tipo "organización". Seguidamente se abre una nueva interfaz donde aparece el formulario donde el usuario debe de llenar todos sus datos y el caso de uso termina cuando realiza la opción guardar dando clic en el botón con el mismo nombre.

#### **Sección a probar en el Caso de Uso**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad



CS.1 Gestionar datos de Organización	EC 1.1: Formulario	El sistema brinda la posibilidad al usuario de llenar sus datos y guárdalos
	EC 1.2 Tratamiento de errores cuando se gestiona datos de organización	El sistema brinda la posibilidad al usuario de avisarle cuando un campo es llenado incorrectamente

### Descripción de Variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Formulario	Campos de textos (TextBox)	No	Se debe de llenar todos los datos del formulario correctamente

### Matriz de Datos

Escenario	Variable de Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Gestionar datos de Organización	T	El sistema brinda la posibilidad al usuario de gestionar su datos	Satisfactorio	<p>Paso 1: El sistema muestra el formulario con todos los campo a llenar</p> <p>Paso 3: El usuario llena los campos pedidos y realiza la opción guardar</p>

Tratamiento de errores cuando se gestiona datos de organización	N	El sistema brinda la posibilidad de avisarle al usuario cuando ha entrado algún campo incorrecto	Satisfactorio	<p>Paso 1: El usuario llena el formulario.</p> <p>Paso 2: El usuario llena el formulario con datos incorrectos.</p> <p>Paso 3: El sistema señala el dato que el usuario lleno incorrectamente y lo marca en rojo</p>
---	---	--	---------------	--

### Caso de prueba del Casos de uso Replicar registro de contacto.

#### Descripción general

El caso de uso inicia cuando el usuario ya termino de gestionar su registro de contacto. El registro se muestra en una interfaz donde el usuario puede apreciar su registro. Si desea replicarlo debe de seleccionarlo y el caso de uso termina cuando realiza la opción replicar dando clic en el botón con el mismo nombre.

#### Sección a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
CS.1 Replicar registro de contacto	EC 1.1: Grid Panel	El sistema brinda la posibilidad al usuario de replicar su registro
	EC 1.2 Tratamiento de errores cuando se replica un registro	El sistema brinda la posibilidad al usuario de avisarle cuando no selecciono ningún registro

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	Grid Panel	Fila de datos	No	Se debe seleccionar la fila correctamente

### Matriz de Datos

Escenario	Variable de Localización	Respuesta del Sistema	Resultado de la Prueba	Flujo Central
Replicar registro de contacto	T	El sistema brinda la posibilidad al usuario de replicar sus datos	Satisfactorio	<p>Paso 1: El sistema muestra la fila con los datos del registro</p> <p>Paso 3: El usuario selecciona la fila y realiza la opción replicar</p>
Tratamiento de errores cuando replica un registro	N	El sistema brinda la posibilidad de avisarle al usuario cuando no ha seleccionado ninguna fila	Satisfactorio	<p>Paso 1: El usuario no selecciona el registro</p> <p>Paso 2: El sistema muestra un mensaje de error. Avisándole al usuario que no ha seleccionado el registro a replicar</p>