

**Universidad de las Ciencias Informáticas**  
**Facultad 6**



**Módulos de Administración y Gestión de Información del  
SIG INRH.**

**Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias  
Informáticas.**

**Autor: Cecilia Esther Hernández Espinosa**

**Tutor: Ing. Yordanys Piñeiro Gómez**

**Ciudad de La Habana, 29 de junio de 2011**  
**“Año 53 de la Revolución”**

*Si quieres ser sabio, aprende a interrogar razonablemente, a escuchar con atención, a responder serenamente y a callar cuando no tengas nada que decir.*

*Johann Kaspar Lavater*

## **DECLARACIÓN DE AUTORÍA**

Declaro que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 29 días del mes de junio del año 2011.

Autor:

Tutor:

---

---

Cecilia Esther Hernández Espinosa

Ing. Yordanys Piñeiro Gómez

## **DATOS DE CONTACTO**

**TUTOR:** Ing. Yordanys Piñeiro Gómez

Graduado de Ingeniería en Ciencias Informáticas en Julio del 2007

Profesor de Ingeniería de Software I y II

Dirección: Universidad de las Ciencias Informáticas (UCI), Edificio: 27, Apto: 201

Teléfono Oficina: +53 – 7 – 8372559. Teléfono Apto: +53 – 7 – 835 - 8867 E-mail: [ypineirog@uci.cu](mailto:ypineirog@uci.cu)

### Síntesis del Tutor

Profesión: Profesor de Ingeniería de Software de la Facultad 6 de la UCI

Categoría docente: Instructor

Años de graduado: 3

## DEDICATORIA

*A mis abuelos Juan y Josefa a los cuales les debo todo lo que soy, por sus enseñanzas, su amor incondicional y por siempre haber confiado en mí.*

*A mi madre, la cual amo con toda las fuerzas de mi corazón y que ha sabido ser siempre mi mejor y más fiel amiga.*

*A Covrado por haber sido como mi padre y al cual siempre le estaré eternamente agradecida.*

*A mi padre que a pesar de que no siempre estuvo presente, en los momentos que estuvo me supo dar su apoyo.*

*A mis tíos Juanito, Rudy y especialmente a mis tías Cecilia y Esther por ser además de mis tías, madres.*

*A mis hermanos, y a mis primos hermanos que también son como mis hermanos, a los más pequeños les dejo el ejemplo.*

*A mi novio por quererme, apoyarme y estar ahí cuando más lo necesitaba.*

*A mis amigos por acompañarme en este viaje y hacerlo más ameno.*

## AGRADECIMIENTOS

*A Fidel Castro por crear esta escuela que me dio la oportunidad de hacer mis sueños realidad.*

*A mis abuelos por todo lo que me inculcaron y a los cuales del debo el haber llegado hasta aquí.*

*A mis padres por darme la vida.*

*A toda mi familia por parte de madre por siempre estar pendiente, cuidarme y sobre todo por confiar en mí.*

*A mis amigos de Campechuela por ser geniales y por siempre estar pendiente de cada paso que doy.*

*A mis amigos de la UCA, especialmente a las Kankas, por acompañarme y estar ahí cuando los necesitaba.*

*A mi tutora Yordany que siempre estuvo ahí cuando tenía dudas, cuando el camino se tornaba difícil, por ser perfeccionista con cada detalle y hacerme dar lo mejor de mí en cada momento.*

*A los miembros del tribunal por cada sugerencia para que este trabajo llegara al final.*

*A Rolando y a Joe por ayudarme y dedicarme tiempo cada vez que lo necesitaba.*

*A mis profesores desde primer año hasta quinto, algunos los mejores otros quizás no tan buenos, pero todos al final aportaron en mi formación un granito de arena y a todos les debo el haber llegado hasta aquí.*

*A los muchachos de la FEM y de la UJC porque con ustedes me hice una mejor persona ya que fuimos como una familia en la cual tuvimos que apoyarnos para cumplir tareas de gigantes.*

*A todos aquellos que en algún momento me preguntaron ¿y la tesis? A todos ustedes les debo este triunfo.*

## RESUMEN

El creciente avance que ha mostrado la ciencia y la técnica en los últimos tiempos ha provocado que hoy se viva en un mundo dependiente de sistemas informáticos y que sean varias las instituciones que a nivel mundial se dedican al desarrollo de este tipo de sistemas. En Cuba la Universidad de las Ciencias Informáticas (UCI), es uno de los organismos que se encuentra inmerso en el desarrollo de software y dentro de las premisas del desarrollo de estos, cuenta con la de velar porque solo las personas autorizadas puedan realizar cambios en la información asociada al negocio del sistema.

El presente trabajo de diploma se encarga de desarrollar los Módulos de Administración y Gestión de la Información para el SIG del Instituto Nacional de Recursos Hidráulicos (SIGINRH), los que se responsabilizan de la autenticación de los usuarios, la asignación de roles y la seguridad de la información, con el fin de lograr mantener la integridad, confiabilidad y disponibilidad de la misma, teniendo como objetivo general la implementación de dichos módulos. Para ello se estarán utilizando métodos científicos como el de análisis y síntesis, el histórico-lógico, el causal y el de modelación. Para la documentación técnica que se desprende de la realización de este trabajo de diploma se estará utilizando como metodología de desarrollo de software Rational Unified Process (RUP), como lenguaje de modelado UML y como herramienta CASE Visual Paradigm, lo que permitirá un correcto entendimiento entre el cliente y el equipo de desarrollo. Para la construcción de la solución se estarán utilizando herramientas y tecnologías que utilicen licencia libre.

El desarrollo de esta investigación tiene un aporte muy importante para el proyecto Aplicativos SIG ya que a través de la misma se prevé proporcionarle al proyecto un módulo que garantice la administración lo cual resulta fundamental y otro que se encargue de manejar la veracidad de los datos introducidos por el usuario que interactúan con el SIGINRH.

### **Palabras claves:**

*Autenticación, confiabilidad, disponibilidad, información, integridad, SIG, SIGINRH.*

## ÍNDICE DE FIGURAS.

Ilustración 1 Diagrama de Clases del Dominio.....	35
Ilustración 2 Diagrama de casos de uso del sistema.....	44
Ilustración 3 Modelo de Despliegue. ....	64
Ilustración 4: Vista lógica de la Arquitectura.....	106
Ilustración 5: DCD Autenticar Usuario.....	107
Ilustración 6: DCD Gestionar Usuario.....	108
Ilustración 7: DCD Gestionar Roles.....	108
Ilustración 8: DCD Gestionar Sistema.....	109
Ilustración 9: DCD Gestionar Funcionalidad.....	109
Ilustración 10: DCD Gestionar Acciones.....	110
Ilustración 11: DCD Gestionar Recurso Hidráulico.....	110
Ilustración 12: DER Módulo de Administración. ....	111
Ilustración 13: DER Módulo de Gestión de la Información.....	112
Ilustración 14: Diagrama de Componentes.....	113

## ÍNDICE DE TABLAS.

Tabla 1 Descripción de los actores del sistema.....	44
Tabla 2 Descripción Textual CU Gestionar Usuarios.....	55
Tabla 3: Diseño de Caso de prueba del CU: Autenticar usuario.....	66
Tabla 4: Matriz de datos del CU: Autenticar usuario. ....	67
Tabla 5: Descripción textual del CU Gestionar Roles .....	82
Tabla 6: Descripción de CU Gestionar Sistema. ....	82
Tabla 7: Descripción textual del CU Gestionar Acciones.....	89
Tabla 8: Descripción textual del CU Gestionar Funcionalidad .....	95
Tabla 9: Descripción textual de CU Gestionar Recurso Hidráulico. ....	102
Tabla 10: Descripción textual de CU Autenticar Usuario. ....	105

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
1.1. INTRODUCCIÓN.....	6
1.2. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	6
1.3. OBJETO DE ESTUDIO.....	13
1.4. ANÁLISIS DE SOLUCIONES EXISTENTES.....	15
1.5. CONCLUSIONES PARCIALES.....	18
<b>CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR</b> .....	<b>19</b>
2.1. INTRODUCCIÓN.....	19
2.2. METODOLOGÍA DE DESARROLLO.....	19
2.3. LENGUAJE DE MODELADO.....	22
2.4. HERRAMIENTA CASE.....	22
2.5. LENGUAJE DE PROGRAMACIÓN.....	24
2.6. ENTORNO DE DESARROLLO INTEGRADO (IDE).....	27
2.7. FRAMEWORK.....	28
2.8. SERVIDOR WEB.....	29
2.9. SISTEMA GESTOR DE BASES DE DATOS (SGBD).....	31
2.10. BASE DE DATOS ESPACIAL.....	32
2.11. POSTGIS.....	32
2.12. CONCLUSIONES PARCIALES.....	33
<b>CAPÍTULO 3. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA</b> .....	<b>34</b>
3.1. INTRODUCCIÓN.....	34
3.2. MODELO DE DOMINIO EN EL DESARROLLO DE SOFTWARE.....	34
3.3. REQUISITOS.....	37
3.4. DESCRIPCIÓN DEL SISTEMA PROPUESTO. MODELO DE CASOS DE USO DEL SISTEMA.....	43
3.5. CONCLUSIONES PARCIALES.....	55

<b>CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>57</b>
4.1. INTRODUCCIÓN.....	57
4.2. ARQUITECTURA DE SOFTWARE .....	57
4.3. DISEÑO.....	62
4.4. DISEÑO DE LA BASE DE DATOS.....	63
4.5. MODELO DE DESPLIEGUE .....	63
4.6. MODELO DE IMPLEMENTACIÓN.....	64
4.7. PRUEBAS.....	65
4.8. CONCLUSIONES PARCIALES.....	68
<b>CONCLUSIONES GENERALES.....</b>	<b>69</b>
<b>RECOMENDACIONES.....</b>	<b>70</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>71</b>
<b>BIBLIOGRAFÍA.....</b>	<b>73</b>
<b>ANEXOS .....</b>	<b>75</b>
ANEXO 1 ESPECIFICACIÓN TEXTUAL DE CASO DE Uso.....	75
ANEXO 2 VISTA LÓGICA DE LA ARQUITECTURA .....	106
ANEXO 3 DIAGRAMA DE CLASES DEL DISEÑO.....	107
ANEXO 4 DISEÑO DE BASES DE DATOS.....	111
ANEXO 5 DIAGRAMA DE COMPONENTES.....	113
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>114</b>

## INTRODUCCIÓN

Desde tiempos inmemorables el hombre ha resguardado y protegido con celo sus conocimientos debido a la ventaja y poder que este le produce sobre otros hombres o sociedades. En la antigüedad surgieron las bibliotecas, lugares donde se podía resguardar la información para evitar que otros la obtuvieran, dando así algunas de las primeras muestras de protección de la información. Es por ello que desde los tiempos de la Edad Media solo los que eran autorizados podían asistir a las bibliotecas que estaban ubicadas en los monasterios, por lo que el control de acceso a estos lugares era bastante riguroso. Más tarde con la creación de algunos inventos como la imprenta, el acceso a la información se volvió más fácil para el resto de los hombres.

Con la creación de las computadoras y más tarde el desarrollo de internet obtener información de cualquier cosa que nos rodea es algo tan sencillo como caminar. Es por eso que hoy la mayor parte de la información que consultamos se encuentra disponible en formato digital.

Pero no todo resulta tan fácil como parece, pues aparejado con este gran avance tecnológico ha ido surgiendo otro fenómeno propio del desarrollo de las Tecnologías de la Informática y las Comunicaciones (TICs) y ha sido el que cualquier persona puede dejar información de cualquier cosa y es esta la causa de que hoy no todo lo que se pueda encontrar en el amplio mundo del internet sea verídico.

Es por eso que para empresas de prestigio internacional es vital que la información que brinden esté correcta y que solo personas autorizadas tengan los permisos para modificar, eliminar o adicionar cualquier fuente. Por tanto para estas instituciones la seguridad y disponibilidad de la información que tienen es algo en lo que deben invertir y no dejar descuidado.

En Cuba existen varias instituciones que se dedican a la producción de aplicaciones informáticas para clientes nacionales e internacionales. Para estas estructuras productoras es fundamental que las soluciones informáticas que se desarrollen garanticen que sólo la información sea actualizada por quien se autorice, evitando que personas extrañas a la misma puedan alterar la integridad y consistencia de los datos que puedan ser confidenciales.

Un ejemplo de las instituciones de Cuba que se dedica a la producción de software es la Universidad de las Ciencias Informáticas (UCI) que se ha convertido en uno de los centros de altos estudios más reconocidos del país, no sólo por la calidad de la enseñanza sino que desde sus inicios al calor de la Batalla de Ideas, ha contemplado entre sus principales actividades la producción de software y por consiguiente la seguridad de estos y de la información que se maneja en los mismos.

En la universidad existen varias facultades y cada una de ellas desarrolla dos o tres líneas y estas líneas son denominadas centros de desarrollo de software. La facultad 6 tiene dos centros, el de Tecnología y Gestión de Datos (DATEC) y Geoinformática y Señales Digitales (GEySED). En este último se desarrollan varios productos y unos de los que más se despliegan actualmente son los Sistemas de Información Geográfica (SIG). Según algunos autores un SIG se define como 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 (georeferenciados), para la solución de problemas complejos del manejo y planeamiento territorial”* (Titular, 2005).

Desde hace más de dos años en el departamento de Geoinformática del centro GEySED se ha ido desarrollando sobre esta línea y ya son palpables los resultados que se han obtenido. Son cada día mayores, las peticiones que llegan desde varios clientes, tanto nacionales como internacionales, que plantean la necesidad de tener a su disposición una potente herramienta como un SIG. Herramienta que les permita seleccionar, adicionar, modificar o eliminar cualquier criterio que se desee mapear y poder analizar cómo diferentes factores afectan un modelo o análisis.

El Instituto Nacional de Recursos Hidráulicos (INRH) es uno de los organismos que ha solicitado la implantación de un sistema de este tipo. En el INRH se realizan diferentes análisis sobre el control de los objetivos hidráulicos cubanos y la información que se genera de estas investigaciones se encuentra en las oficinas del centro dispersa y en papeles. Resulta complejo realizar un análisis profundo para tomar alguna decisión y en ocasiones darle solución a algunos problemas requiere de acceso a diversos tipos de información, que solo pueden estar relacionadas territorialmente o por distribución espacial.

Dentro del proyecto Aplicativos-SIG surge el producto SIG-INRH como respuesta a las necesidades planteadas anteriormente, de informatizar todos estos procesos que se llevan a cabo, de una forma u otra, en el INRH. Esta concebido que se desarrolle un SIG que permita recopilar y manejar información usando

datos geográficos, analizando patrones, relaciones, y tendencias en la información; todo con el interés de contribuir a la toma de mejores decisiones.

Para el INRH también es muy importante que solo los especialistas calificados puedan realizar determinadas acciones sobre el sistema y que solo para los que se decida estén visibles unas u otras funcionalidades del SIG, por lo que se hace necesario controlar los accesos y que solo el personal autorizado pueda realizar determinados cambios. De acuerdo con la situación expuesta se puede definir que el **problema a resolver es**: ¿Cómo contribuir a la gestión de la información correspondiente al control de objetivos hidráulicos cubanos que se lleva a cabo en el Instituto Nacional de Recursos Hidráulicos así como a la seguridad del SIG de esta Institución?

Con el fin de resolver el problema descrito se hace necesario implementar los subsistemas de Gestión de la Información y Administración de usuarios correspondiente al SIG del INRH, siendo este el **objetivo general** de la investigación.

Para darle cumplimiento al objetivo general fue necesario centrar la investigación en los procesos de gestión de información y administración de los SIG quedando como **objeto de estudio** de la investigación.

El **campo de acción** está enmarcado en la administración y gestión de la información de los SIG aplicados a los objetivos hidráulicos cubanos.

Para darle cumplimiento al objetivo general referido se trazaron las siguientes **tareas de la investigación**:

1. Caracterizar los procesos que se llevan a cabo en el INRH para el control de objetivos hidráulicos cubanos.
2. Analizar los conceptos para conocer las inquietudes del cliente y lograr una familiarización con estos.
3. Argumentar el uso de la metodología de desarrollo a utilizar, lenguaje de modelado y herramientas CASE.
4. Elaborar la documentación técnica correspondiente al Modelo de Dominio para lograr familiarizarse con el entorno de trabajo que actualmente existe.
5. Describir los requisitos que deben cumplir los subsistemas.
6. Elaborar la documentación técnica correspondiente al Modelo de Casos de Uso de los Subsistemas.
7. Elaborar la documentación técnica correspondiente al Modelo de diseño de los Subsistemas.
8. Elaborar la documentación técnica correspondiente al Modelo de Implementación de los Subsistemas.

9. Elaborar la documentación técnica correspondiente al Modelo de Despliegue de los Subsistemas.
10. Implementar los Subsistemas de Administración y Gestión de Información del Sistema de Información Geográfico para el control de objetivos hidráulicos cubanos.
11. Probar el sistema respecto al cumplimiento de los requisitos planteados.

El desarrollo satisfactorio de las tareas expuestas anteriormente contribuirá al desarrollo exitoso de la **idea a defender**: la implementación de los Módulos de Administración y Gestión de Información del SIG para el control de objetivos hidráulicos cubanos, permitirá controlar el acceso de los usuarios a la aplicación y que la información asociada a cualquier recurso a representar sea certera.

Para el desarrollo de la misma se decidió recurrir a los métodos tanto teóricos como empíricos. De los teóricos se utilizaron el histórico-lógico, análisis-síntesis, causal y el de modelación. De los empíricos solo las entrevistas.

*El método Histórico-Lógico se aplica para investigar la existencia de algún sistema que se utilice para el control de los recursos hidráulicos cubanos.*

El método de Análisis y Síntesis, se define con el objetivo de analizar los diversos documentos de los procesos de administración de usuarios y gestión de la información de los SIG así como para realizar una síntesis de los mismos.

El método causal se emplea para determinar y analizar los factores que provocan la necesidad de desarrollo de un SIG para el INRH.

El método de Modelación se emplea para mostrar los diferentes artefactos que se construyen como resultado del proceso de ingenieril.

Las entrevistas se realizarán a los clientes del INRH y a especialistas con experiencia en el desarrollo de SIG para recopilar la información referente a la seguridad de la administración de los usuarios y a la gestión de la información de los SIG aplicados a los objetivos hidráulicos.

Los posibles resultados del trabajo de diploma están dados en la implementación del subsistema de gestión de la información, en la implementación del subsistema de administración correspondiente al SIG para el control de los objetivos hidráulicos cubanos y en la documentación técnica del proceso ingenieril de los dos subsistemas a implementar. El presente documento se estructura en cuatro capítulos que a continuación se describen:

El **Capítulo 1** se enmarca en la fundamentación teórica de la investigación, en el **cual** serán expuestos los principales conceptos asociados al problema en cuestión para lograr un mayor entendimiento por parte de los usuarios ajenos al sistema. Se explican además todos los argumentos del objeto de estudio así como una síntesis de algunas soluciones existentes para el problema identificado.

En el **Capítulo 2** se describen los tipos de metodologías, lenguaje de modelado y herramientas CASE que son las más adecuadas para la construcción de los artefactos correspondientes a cada uno de los flujos de trabajo.

Por su parte en el **Capítulo 3** se definen los requisitos funcionales y no funcionales del sistema, se identifican los casos de uso del sistema, se realizan las descripciones textuales de los mismos y se estructura el Modelo de Casos de Uso del Sistema.

Por último en el **Capítulo 4** se definen las clases del diseño, sus relaciones y se estructuran los diagramas de secuencia del diseño. Se realiza además el Modelo de Implementación y de Despliegue. Se implementarán cada una de las funcionalidades de forma iterativa e incremental hasta lograr el producto completo. Finalmente se realizan las pruebas de caja negra utilizando la técnica de partición de equivalencia.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

El simple hecho de tener un SIG para facilitar el trabajo en una institución hace que la seguridad y la manipulación de los datos sean tan exigentes como tan sensible sea la información que se maneje. Es por ello que la administración de usuarios y la gestión de la información del SIG INRH son un eslabón priorizado.

En el presente capítulo se realiza un análisis del objeto de estudio, para caracterizar la administración y gestión de la información dentro de los sistemas informáticos. Contiene además los principales aspectos que conforman la fundamentación teórica de la investigación y que crean una visión de la misma, entre ellos los principales conceptos asociados al dominio del problema y la descripción de la situación problemática. Se analizan algunas soluciones similares para obtener una perspectiva de la utilización de funcionalidades afines a las que se proponen en la presente investigación.

### 1.2. Conceptos Asociados al dominio del problema

Con el propósito de que el lector pueda alcanzar una mejor comprensión de los temas que serán tratados en este capítulo, directamente relacionados con el objeto de estudio de la investigación, se describen a continuación un grupo de conceptos coligados al dominio del problema, entre los que se destacan: Control de acceso, Identificación, Autenticación, Autorización, Personalización, Auditoría, Confidencialidad, Integridad, Disponibilidad, Sistema Informático, Sistema de Información Geográfica, Datos Espaciales, Entorno Libre o Software Libre.

#### **Control de Acceso**

Es el proceso de conceder permisos a usuarios de acceder a objetos, información, datos como ficheros o impresoras en la red. Está basado en tres conceptos fundamentales: identificación, autenticación y autorización. Incluye autenticar la identidad de los usuarios y autorizar el acceso a datos o recursos. Son necesarios para proteger la confidencialidad, integridad y disponibilidad de los objetos, y por extensión, de la información que contienen, pues permiten que los usuarios autorizados accedan solo a los recursos que ellos requieren para realizar sus tareas. (Sotelo 2008)

## **Identificación**

Es la acción por parte de un usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario. Establece además, que el usuario es responsable de las acciones que lleve a cabo en el sistema. Esto está relacionado con los registros de auditorías que permiten guardar las acciones realizadas dentro del sistema y rastrearlas hasta el usuario autenticado. (Sotelo 2008)

## **Autenticación**

Es el proceso de descubrir y comprobar la identidad de un usuario mediante el examen de sus credenciales y la validación de las mismas consultando a una autoridad determinada. Actualmente se utilizan una gran variedad de mecanismos de autenticación. (Sotelo 2008)

## **Autorización**

Es el proceso que determina si el usuario o proceso previamente identificado y autenticado tiene permitido el acceso a los recursos. (Sotelo 2008)

## **Personalización**

Es un proceso en el cual la aplicación asume la identidad del invocador y que, en su nombre, realice solicitudes a otros recursos. El acceso se concede o deniega en función de la identidad que se adopte. Si esta identidad dispone de permisos para un recurso, la aplicación que la adopta también dispondrá de los mismos permisos de acceso. (Sotelo 2008)

## **Auditoría**

Es un registro cronológico de los eventos relevantes a la seguridad de un sistema. Este registro puede luego examinarse para reconstruir un escenario en particular. (Sotelo 2008)

## **Confidencialidad**

La información o los activos informáticos son accedidos solo por las personas autorizadas. Sería fácil mantener la confidencialidad de un sistema si nadie tuviera acceso a él, pero este tendría una disponibilidad nula. (Sotelo 2008)

## Integridad

La información que está en correspondencia con la realidad solo puede ser modificada por las personas autorizadas y de la forma autorizada. (Sotelo 2008)

## Disponibilidad

La información solo puede ser accedida por las personas autorizadas en el momento requerido. (Sotelo 2008)

## Sistema de Información Geográfica

Un **Sistema de Información Geográfica** (**SIG** o **GIS**, en su acrónimo inglés [*Geographic Information System*]) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de **resolver problemas complejos de planificación y gestión**. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres y construidas para satisfacer unas necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. (Chang 2006)

## Técnicas de Autenticación

Para la autenticación de usuarios de los SIG se hace una descripción de las principales técnicas que existen.

- Sistemas basados en objetos de seguridad.
- Sistemas de autenticación biométrica.
- Sistemas de autenticación por certificados.
- Sistemas basados en contraseñas.

A continuación se describe cada una de ellas:

### ***Sistemas basados en objetos de seguridad***

Distinto a una contraseña, un objeto de seguridad es un objeto físico, o sea, un pequeño dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio de red o una aplicación. El dispositivo puede ser en forma de una tarjeta inteligente o puede estar incorporado en un objeto utilizado comúnmente, como un llavero. (IRIS-CERT, 2008)

### ***Sistemas de autenticación biométrica***

Con el acelerado avance que hoy muestra la ciencia y la técnica hace que se pueda decir que los sistemas de autenticación que lograrán imponerse en un futuro no muy lejano serán los de autenticación biométrica, pues serán los más robustos cuando se haga necesario autenticar un usuario. Estos sistemas son más amigables para el usuario, no se necesita recordar contraseñas o números de identificación complejos, y como se suele decir, el usuario puede olvidar una tarjeta de identificación en casa, pero nunca olvidará una parte de su cuerpo. Son mucho más difíciles de falsificar que una simple contraseña o una tarjeta magnética; las principales razones por la que no se han impuesto ya, es su elevado precio, fuera del alcance de muchas organizaciones, y su dificultad de mantenimiento.

Los sistemas antes mencionados son los denominados biométricos, que se basan en rasgos personales distintivos con capacidad de identificar a una persona. Se clasifican en: (IRIS-CERT, 2008)

- *Fisiológicos*: huella dactilar, iris, retina, cara, geometría de la mano, huella palmar, estructura de las venas, estructura de la oreja, termografía facial.
- *Conductuales*: voz, escritura, firma manuscrita, modo de teclear, modo de andar.

### ***Sistemas de autenticación por certificados***

Los sistemas de autenticación por certificados además de autenticar al servidor frente a los clientes, también es posible que los clientes se identifiquen mediante la presentación de un certificado. Los certificados de cliente normalmente contienen información como su nombre, empresa, departamento, dirección de correo, ciudad, país, etc., lo que permite sofisticados esquemas de control de acceso basándose en uno de estos atributos o en una combinación de varios o en el conjunto del certificado. (Marañón 2002)

### ***Sistemas basados en contraseñas***

Una contraseña o clave es una forma de autenticación que utiliza información que solamente el individuo conoce, o sea secreta, para controlar el acceso hacia algún recurso protegido. Está compuesta por un código alfanumérico y en ocasiones solamente numérico. Mientras que las contraseñas crean una seguridad contra los usuarios no autorizados, el sistema de seguridad sólo puede confirmar que la contraseña es válida, y no si el usuario está autorizado a utilizar esa contraseña. Esa es la razón por la que las contraseñas normalmente deben mantenerse en secreto ante aquellos a quien no se le permite el acceso. (IRIS-CERT, 2008)

Como en todas las técnicas de autenticación, existe una relación entre seguridad y conveniencia (facilidad de uso). Es decir, si algún recurso está protegido por una contraseña, entonces la seguridad se incrementa con la consecuente pérdida de conveniencia para los usuarios. No existe un método que sea el mejor para definir un balance apropiado entre seguridad y conveniencia.

- La seguridad de las contraseñas se ve afectada por diversos factores que se mencionarán a continuación:(IRIS-CERT, 2008)
- *Fortaleza de la contraseña:* Deben ser largas, normalmente más de siete caracteres, y se deben usar combinaciones de letras mayúsculas y minúsculas, números y símbolos. Ejemplos de contraseñas fuertes serían las siguientes: tastY=wheeT34, pArtei@34! y #23kLLflux.
- *Formas de almacenar las contraseñas:* Se debe usar un algoritmo criptográfico irreversible (o función resumen), los más comunes son SHA1 y SHA2.
- *Método de retransmisión de la contraseña:* Deben ser transmitidas mediante algún método criptográfico, en el caso de las redes locales se usa con mucha frecuencia Kerberos.
- *Longevidad de la contraseña:* Deben ser cambiadas cada 60 días.

Algunos de sus inconvenientes son:

- Los usuarios no suelen elegir contraseñas especialmente robustas, problema que constituye una de las mayores amenazas a la seguridad de un servidor, cuando se trata de contraseñas de una cuenta, o a la confidencialidad de los datos, cuando se trata de un acceso a web.
- Los usuarios pueden revelar sus contraseñas a otras personas, que podrían así, acceder también a los servicios protegidos.

- Los usuarios suelen perder u olvidar sus contraseñas, requiriendo ayuda por parte de los administradores para resolver el problema.

Se llega a la conclusión de utilizar los sistemas basados en contraseña pues las ventajas de utilizar este tipo de autenticación es que es sencillo, barato, y está altamente probado. Es sencillo porque el usuario no tiene que utilizar ningún dispositivo externo para autenticarse en el sistema. Es barato debido a que no se necesita de un coste adicional a la hora de entrar a un sistema y poner su contraseña, pero en cambio, los sistemas que utilizan objetos para poder autenticarse, necesitan de un gasto extra para utilizar un identificador de objetos. Los sistemas biométricos son mucho más novedosos y se están desarrollando a gran velocidad, aunque tiene desventajas que retardan su desarrollo como el precio de los equipos de captación, conceptos éticos y poca costumbre de utilización. Está probado que los sistemas basados en contraseñas son los más utilizados en el mundo, porque con diferentes variantes son aplicados en casi todos los aspectos de la seguridad de la información.

Igualmente todos los sistemas se pueden combinar para aumentar la seguridad, especialmente el uso de contraseñas normalmente acompaña a los sistemas biométricos y los de objetos.

## Medidas de Seguridad

Diseñar un sistema seguro es una tarea muy complicada pues la creciente complejidad de los sistemas informáticos, de los que se demandan más movilidad, flexibilidad y conectividad, ha llevado al crecimiento exponencial de las amenazas. Implantar medidas de seguridad para proteger sistemas informáticos conlleva un proceso de análisis de las vulnerabilidades del sistema. La definición de una política de seguridad y su implementación, a través de una serie de medidas y mecanismos de defensa, obliga a subordinar algunas de las ventajas del sistema informático. Se hace obvio que a mayores y más estrictas medidas de seguridad, menos amigable es el sistema. Estas medidas de seguridad pueden ser de tres tipos fundamentales: físicas, lógicas y administrativas. (Sotelo 2008)

Medidas físicas: Se aplican mecanismos para impedir el acceso directo o físico no autorizado al sistema. También protegen al sistema de desastres naturales o condiciones medioambientales adversas. Existen tres factores fundamentales a considerar:

- El acceso físico al sistema por parte de personas no autorizadas.

- Los daños físicos por parte de agentes nocivos o contingencias.
- Las medidas de recuperación en caso de fallo.

Medidas lógicas: incluye un conjunto de medidas de acceso lógico a los recursos y a la información, garantizando el uso correcto de los mismos. Se refiere más a la protección de la información almacenada. Entre los tipos de controles lógicos que son **posibles** incluir en el sistema de seguridad se destacan los siguientes:

Implementación de un sistema de control de accesos a los recursos y la información.

- Uso de la criptografía para proteger los datos y las comunicaciones.
- Empleo de cortafuegos para proteger una red local de Internet.
- Realización de copias de respaldo de la información.
- Monitoreo y auditoría del sistema.

*Medidas administrativas:* Son aquellas que deben ser tomadas por las personas encargadas de definir la política de seguridad de la institución para ponerla en práctica, hacerla viable y vigilar su correcto funcionamiento. Algunas de las medidas administrativas fundamentales a tomar son las siguientes:

- Documentación y publicación de la política de seguridad de la institución y de las medidas tomadas para ponerla en práctica.
- Establecimiento de un plan de formación del personal.

## Seguridad de la Información

Se entiende por seguridad de la información a todas aquellas medidas preventivas y reactivas del hombre, de las organizaciones y de los sistemas tecnológicos que permitan resguardar y proteger la información buscando mantener la confidencialidad, la autenticidad e integridad de la misma. (ISO/27000 2009)

El concepto de seguridad de la información no debe ser confundido con el de seguridad informática, ya que este último sólo se encarga de la seguridad en el medio informático, pudiendo encontrar información en diferentes medios o formas.

Para el hombre como individuo, la seguridad de la información tiene un efecto significativo respecto a su privacidad, la que puede cobrar distintas dimensiones dependiendo de la cultura del mismo. El campo de la seguridad de la información ha crecido y evolucionado considerablemente a partir de la Segunda Guerra Mundial, convirtiéndose en una carrera acreditada a nivel mundial.

En la seguridad de la información es importante señalar que su manejo está basado en la tecnología y se debe conocer que puede ser confidencial: la información está centralizada y puede tener un alto valor. Puede ser divulgada, mal utilizada, robada, borrada o sabotada. Esto afecta su disponibilidad y la pone en riesgo.

Dentro de los SIG es muy importante mantener esta seguridad sobre la información que se muestra pues sería muy engorroso que alguien no autorizado manipulara la información respecto a la ubicación geográfica de cualquier objeto.

### **1.3. Objeto de Estudio**

Con el paso de los días la conexión de las computadoras se hace superior y la tendencia es hacia un aumento acelerado, las ventajas que trae son incalculables, lo que obliga a prestar especial atención a la administración y la gestión de la información que se intercambia.

El control de acceso a los SIG definido por roles es una de las cuestiones más importantes que se manejan con respecto a la seguridad y sin lugar a dudas esto permite que solo las personas con la autorización requerida puedan hacer cambios o modificar cualquier fuente. En nuestros días los ataques piratas a través de la red hacia sistemas de este tipo son más frecuentes y sus efectos más devastadores, por tal motivo, se hace de vital importancia tomar las medidas adecuadas para evitar fugas o pérdidas de datos, ya sean accidentales o intencionados, en todos los niveles de desarrollo, implantación y mantenimiento.

### **1.3.1. Descripción actual del dominio del problema**

El Instituto Nacional de Recursos Hidráulicos es el organismo encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en cuanto a las actividades de los recursos hidráulicos del país. (Hidráulicos 1989).

Fue fundado el 6 de junio de 1989 y tiene asociadas varias entidades como la Empresa Mixta Aguas de la Habana, La Asociación Internacional Aguas Varadero y la Empresa Cubahidráulicos.

En el INRH diariamente se actualizan los datos de cantidad de m<sup>3</sup> de los ríos, embalses y presas y con cierta frecuencia se hacen ciertos análisis sobre en qué lugar ubicar uno, se determinan las distancias de uno a otro, se hallan caminos así como otros estudios. Todas estas observaciones se guardan en un archivo físico para una posterior consulta.

Como el INRH no cuenta con un sistema que les permita realizar consulta de todos los datos en tiempo real se hace complejo tener en un corto período de tiempo la información más actualizada posible de cualquier recurso hidráulico.

### **1.3.2. Situación Problemática**

En Cuba el INRH se creó con el objetivo de que una institución se encargara de la administración y control de todos los objetivos hidráulicos cubanos, para mantener centralizados todos los datos referentes a estos, mantener un control estricto de las zonas más vulnerables a las sequías y ubicar los puntos estratégicos para la construcción de los mismos.

El INRH es la organización que garantiza el agua de toda la isla con fines económicos y sociales. Tienen la responsabilidad de mantener el control estricto de estos objetivos y que se les de el correcto uso.

Para los trabajadores del INRH resulta muy complejo realizar análisis de diferentes fuentes a la vez, porque se encuentran ubicadas en diferentes archivos en los cuales resulta complejo realizar búsquedas y comparar criterios.

Los especialistas del instituto con frecuencia deben informar la situación actual de los diferentes embalses ubicados en el territorio cubano, que de tener un sistema que tenga esta información ubicada geográficamente solo sería entrar a una aplicación y actualizar los datos necesarios.

A este organismo se le desarrolla un sistema que le permitirá el control, georreferenciación y análisis de los embalses del país, que permita la toma de decisiones sobre la situación hidrológica actual y la actualización inmediata de los datos referentes a estos objetivos.

El sistema que se desarrolla no cuenta con un módulo que garantice la administración y la actualización de los datos solo por los usuarios autorizados, por lo que se necesita que solo algunos estén autorizados a realizar ciertas operaciones. A otros se les permitirá un acceso total a todos los datos, mientras que a otros sólo se les permitirá acceso a un subconjunto de los datos y otros tendrán acceso de sólo lectura.

Dado que los SIG son un poderoso conjunto de herramientas para almacenar, recuperar, transformar y exhibir datos espaciales referenciados al mundo real, resulta muy importante que la información que brinden sea certera. Resultando la veracidad de la información su eslabón más fuerte y al mismo tiempo el más débil porque día a día hackers y atacantes informáticos tratan de interrumpir el correcto funcionamiento de cualquier aplicación informática.

La actualización, creación, aprobación, distribución y consulta de un documento o registro, solo pueden ser manejados, por el personal que tenga permisos para hacerlo; el correcto funcionamiento del SIG INRH se garantiza con el cumplimiento de las premisas antes expuestas.

#### **1.4. Análisis de soluciones existentes**

Profesionales de la seguridad hoy en día se enfrentan a la pérdida de datos de aplicaciones web y servicios web de gestión de los riesgos de seguridad. Necesitan un sistema informatizado, programado, dinámico, continuo y de aplicación de análisis de seguridad, con el objetivo de facilitar recursos que capaciten a los desarrolladores de sistemas informáticos a afrontar dichos retos, y de solucionar los problemas de este tipo en las aplicaciones. Varios productos y sistemas se han desarrollado a nivel mundial y aunque no cubren en su totalidad la diversidad de obstáculos que de este género van surgiendo, brindan grandes beneficios en cuestiones de protección de datos y sistemas.

La investigación que se realiza está dividida en dos aristas: una es la administración del sistema y otra la gestión de la información que se brinda, por lo que el análisis de las soluciones existentes será realizada hacia estas dos líneas. Una aplicación en un entorno web será la solución que se propone por lo que se hace necesario realizar un estudio de los sistemas que existen en el mundo, en Cuba y en la UCI para garantizar la administración y la gestión de la información de aplicaciones web.

## **Single Sign-On (SSO)**

Tal como indica el significado del término, Single Sign-On hace referencia a la posibilidad de que los usuarios de sistemas informáticos realicen por única vez el procedimiento de identificación y autenticación para el acceso a los diferentes servicios brindados dentro de la infraestructura informática local. El mecanismo habitual para lograr esta funcionalidad es que el procedimiento de identificación y autenticación dé como resultado un conjunto de credenciales que pueden ser posteriormente utilizadas para demostrar la identidad de los usuarios en el acceso a los diferentes servicios, sin necesidad de volver a proporcionar la información de autenticación. La implementación de un sistema de este tipo aporta una clara ventaja y es que las contraseñas no transitan de forma insegura ya que los usuarios solo necesitan recordar una sola contraseña y no una para cada acceso que necesiten hacer. Garantiza el poder establecer, mediante políticas de control de acceso, qué usuarios pueden acceder a qué aplicaciones y recursos, de manera que un usuario no pueda utilizar funcionalidades para las que no está autorizado.

Existen muchas soluciones de SSO en el mercado, pero algunos aspectos definen el que una solución SSO concreta tenga éxito en proyectos reales: el SSO no debe requerir modificaciones ni en las aplicaciones ni en los clientes para acceder a ella, sino que debería ser puramente plug & play. Las modificaciones en las aplicaciones o en los clientes son caras, requieren tiempo y no siempre son posibles. La modificación en los procedimientos de trabajo de los usuarios (lanzaderas o iconos especiales, por ejemplo), no contará con su aceptación, que es, en muchos casos, el elemento más crítico de un proyecto de SSO. Además, puede no ser viable por precio, tiempo o limitaciones puramente técnicas.

## **Sistema de Gestión Integral de Seguridad (SGIS)**

El Sistema de Gestión Integral fue desarrollado en la UCI en el año 2009 por el Centro UCID y tiene como objetivo brindar sus servicios a todos los sistemas que se suscriban a él. Para ello se gestionan las conexiones a la base de datos, las funcionalidades asociadas y las acciones que realizan las mismas. Una vez registrada esta información se procede a la creación de roles a los cuales se les dan los permisos dentro de cada sistema. Luego se crean los usuarios con el perfil seleccionado y se le asignan uno o muchos roles en una o muchas entidades respectivamente.

## GIS Mapping

Es un componente clave para la gestión de recursos naturales. Este SIG se utiliza para modelar y administrar las poblaciones de vida silvestre, el crecimiento de la madera y la cosecha, y la gestión de la calidad del medio ambiente. Este SIG es una herramienta importante para identificar las fuentes de contaminación y las soluciones que cumplen con las leyes estatales y federales de medio ambiente. Un ejemplo es la utilización del Condado de Sacramento de la cartografía SIG para identificar las fuentes no puntuales de contaminación del agua, los tipos de uso de la tierra que contribuyen al máximo a la contaminación, así como una evaluación de la planificación de escenarios relacionados con la mitigación y reducción de dichas fuentes.

### Observación de las soluciones existentes

El estudio de Sistemas de Seguridad a nivel internacional y nacional permitió identificar el conjunto de funcionalidades básicas que debe poseer la aplicación que se desea construir. SSO es un sistema que brinda grandes ventajas en cuanto a los accesos y permisos a aplicaciones, pero como bien se describe resulta elevada la adquisición de su uso; Cuba se encuentra en una migración a Software Libre y especialmente la UCI aumenta sus esfuerzos en lograr una soberanía tecnológica con el uso de tecnologías libres y este tipo de sistema no garantiza el cumplimiento de esta política. Otro ejemplo claro de las aplicaciones SSO es la plataforma v-Go Single ON, producida por la compañía de Passlogix, tiene precios elevados, la distribución de esta plataforma en conjunto con IBM tiene un costo de 75 USD por usuario que incluye un valor de licencia única y su mantenimiento por un año; si se compra su versión multilingüe y multiplataforma que incluye el CD-ROM de instalación asciende a un costo de 140 USD por usuario y los servicios de instalación y configuración tendrán un valor de 3.60 USD por usuario.

En la UCI se han desarrollado varios sistemas de seguridad pero el más completo es el SGIS que permite realizar una administración centralizada de las aplicaciones pero su uso resulta engorroso para soluciones pequeñas, puesto que está desarrollada para administrar grandes proyectos. Dado que las soluciones encontradas tanto nacional como internacional no tiene integrados todos los requisitos funcionales que se necesitan; se decidió implementar el subsistema de seguridad para el SIGINRH.

A nivel internacional existen varias soluciones SIG que se utilizan para gestionar recursos hidráulicos, pero esta gestión la realizan a través de los datos que se revelan de la cartografía que brinda un servidor de

mapas y no desde un Sistema Gestor Base de Datos Espaciales. Esta característica no permite que los datos referentes a la ubicación geográfica puedan ser tratados por especialistas. En Cuba no existe un sistema que permita la gestión de recursos hidráulicos desde una visión territorial. El estudio de estas soluciones ayudaron a identificar un conjunto de atributos y funcionalidades que no deben faltar pero como no permiten que la gestión de los recursos hidráulicos pueda ser tramitada por los especialistas se decidió implementar un subsistema que permita la gestión de los recursos hidráulicos del INRH.

## **1.5. Conclusiones Parciales**

La administración y gestión de la información garantizan el correcto funcionamiento de aplicaciones informáticas ya que los ataques hoy en día son cada vez más sofisticados y resulta más fácil interrumpir el correcto funcionamiento de las aplicaciones que se encuentren disponibles. Un SIG resulta una mina de oro para la empresa que lo utilice para la gestión de sus medios y servicios desde una visión territorial, por lo que estos sistemas deben estar doblemente protegidos, por la importancia que tiene la información que ofrecen ya que cambios en ella mal intencionados podrían causar grandes pérdidas. A lo largo de todo el capítulo se han caracterizado los conceptos asociados al tema, los cuales ayudan a una mejor comprensión del objeto de estudio; al mismo tiempo se ha realizado un análisis detallado de los factores que influyen en la situación problemática. Con el análisis de las diferentes técnicas de autenticación se pudo concluir que el sistema de autenticación más factible a utilizar es el basado en contraseña por las características y ventajas antes expuestas.

Otro rasgo significativo es que se han expuesto las posibles soluciones al problema, realizando un profundo estudio de los principales sistemas de administración y gestión existentes, el cual arrojó como resultado que ninguna de las referenciadas cumplen con los requisitos específicos que persigue el SIG INRH.

## CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR

### 2.1. Introducción

Mantener la información fuera del alcance de usuarios no autorizados resulta fundamental para el correcto funcionamiento de las soluciones informáticas que se brindan a los clientes que las soliciten. Cada día los ataques informáticos con el fin de garantizar que los recursos informáticos de una compañía no estén disponibles para cumplir sus propósitos, es decir, que estén dañados o alterados por circunstancias o factores externos se hacen más fuertes. Desarrollar aplicaciones que no cuenten con un subsistema integrado que se encargue de la administración y gestión de la información es un verdadero caos.

Dada la importancia que tienen estos subsistemas integrados al SIG que se desarrolla es que en el presente capítulo se expone el resultado de la investigación sobre la metodología de desarrollo, lenguaje y herramientas que ayudarán a construir la solución solicitada por el cliente.

### 2.2. Metodología de desarrollo

Definir una metodología para el desarrollo del software ayuda a que este sea menos complejo y que guíe de manera coherente sus actividades durante el proceso de desarrollo del producto.

Por lo general, si se quiere construir un software de alta calidad, desarrollado en el tiempo planificado y con los costes establecidos; pero que además satisfaga la necesidad de ser elaborado de una forma más acelerada es necesario enfocarse en trabajar de forma organizada, donde se controle y documente todo lo relacionado con el proyecto en cuestión y puedan eliminarse los riesgos que podrían presentarse durante el desarrollo del mismo; lo cual no podría lograrse sin el empleo de una metodología eficaz que se adapte a las características propias del software que se esté desarrollando.

“Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas)

actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y/o diseño”.  
(VALENCIA)

Clasificar las metodologías es una tarea bien difícil por la gran cantidad de propuestas y diferencias en el grado de detalle, información disponible y alcance que poseen. A grandes rasgos, considerando su filosofía de desarrollo se pueden agrupar en: metodologías ágiles o ligeras y metodologías pesadas o tradicionales.

Las metodologías tradicionales o pesadas se guían expresamente por una fuerte planificación durante todo el proceso de desarrollo, en el cual se establecen estrictamente las actividades involucradas, los roles definidos, los artefactos que se deben producir, las herramientas y notaciones que serán utilizadas así como el modelado y documentación detallada.

Las metodologías ágiles se enfocan en evadir las burocráticas guías de las metodologías tradicionales. Centran sus procesos en heurísticas provenientes de prácticas de producción de código, donde el cliente es parte del equipo, con ciclos muy cortos de desarrollo y preparadas para cambios durante el proyecto. Los grupos de trabajo están diseñados con poco personal e interactúan en el mismo sitio, generan pocos artefactos y poseen pocos roles.

Después de realizar un estudio detallado de un conjunto de metodologías se decidió el uso de una metodología tradicional porque se necesita la existencia de una documentación detallada de todo el proceso de desarrollo, para que pueda ser reutilizada en futuras versiones o en el desarrollo de sistemas similares. También, se caracterizan por centrar su atención en cumplir con un plan de proyecto definido en la fase inicial. Intuye un equipo de desarrollo numeroso dividido por roles y el cliente no forma parte del equipo de desarrollo. Con respecto a la curva de aprendizaje, los modelos ágiles, ofrecen una mayor ventaja pero con ciertas limitaciones, ya que aún no han sido explotadas a gran escala como sucede con Rational Unified Process (RUP metodología tradicional) que posee alto soporte y herramientas integrales que guían al equipo de desarrollo, facilitando aplicar con mayor efectividad esta metodología, permitiendo aprovecharla al máximo, es por eso que se decide el uso de RUP.

### **2.2.1. Rational Unified Process (RUP)**

El Proceso Unificado Racional o de Desarrollo (Rational Unified Process en inglés, tradicionalmente abreviado como RUP) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es el resultado de varios años de trabajo y uso práctico en

el que se han unificado técnicas de desarrollo. Utiliza como lenguaje de modelado el UML (Lenguaje Unificado de Modelado). Define como sus principales elementos: Trabajadores (“quién”), Actividades (“cómo”), Artefactos (“qué”), Flujo de actividades (“Cuándo”).

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo, los 6 primeros conocidos como flujos de ingeniería (*Modelamiento del Negocio, Requisitos, Análisis y Diseño, Implementación, Prueba e Instalación*) y los tres últimos de apoyo (*Administración del proyecto, Administración de configuración y cambios y Ambiente*). Cada flujo de trabajo cumple con algunas actividades específicas. El Proceso Unificado de Desarrollo posee trabajadores específicos que producen y consumen artefactos también definidos. Cada fase representa un estado del proyecto, y produce un hito que sirve de entrada a la próxima fase.

El ciclo de vida de RUP se caracteriza por ser dirigido por casos de uso siendo los casos de uso los que reflejan lo que los usuarios futuros necesitan y desean, guiando el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo representan la realización de los casos de uso. Además, es centrado en la arquitectura ya que muestra la visión común del sistema completo y describe los elementos del modelo que son más importantes para su construcción. También es iterativo e incremental, que significa que una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros, resultando éstas en un incremento conocido como miniproyecto. Con estos miniproyectos se va obteniendo un producto con un determinado nivel que irá creciendo incrementalmente en cada iteración. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

Para el desarrollo de los subsistemas de gestión de la información y administración del SIG-INRH, se definió el uso de RUP como metodología porque permite de forma disciplinada asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo). Además, es el proceso de desarrollo más general de los existentes actualmente, es adaptable a la mayoría de los proyectos de desarrollo del software; característica que lo ha convertido en una de los más utilizados actualmente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML. Facilita el desarrollo del proyecto porque divide el trabajo en partes más pequeñas o miniproyectos. Hace que la programación sea más sencilla. Garantiza la reutilización. Provee al equipo de la documentación detallada de todo el proceso de desarrollo.

## 2.3. Lenguaje de Modelado

Un lenguaje para modelar objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos. Muchas organizaciones lo usan en combinación de una metodología de desarrollo para avanzar de una especificación inicial a un plan de implementación para todo el equipo de desarrolladores. Luego de la conclusión de la metodología a utilizar indudablemente el lenguaje de modelado por el que se va a guiar el desarrollo del software es el Lenguaje Unificado de Modelado (UML).

### 2.3.1. UML

Se utiliza este lenguaje de modelado ya que permite especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener, y controlar la información sobre tales sistemas. Admite realizar una verificación y validación del modelo realizado. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. También ofrece nueve diagramas en los cuales modelar sistemas. Permite automatizar determinados procesos y soporta generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos), lo cual garantiza que el modelo y el código estén actualizados, manteniendo la visión en el diseño de más alto nivel, de la estructura de un proyecto. Además, este lenguaje de modelado es el que utiliza la metodología seleccionada para la solución.

## 2.4. Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, dado el diseño se puede obtener automáticamente la implementación de parte del código, compilación automática, documentación o detección de errores entre otras.

Una herramienta CASE se utiliza para ayudar a las actividades del proceso de software que es utilizado para diseñar y para implementar otro software. Facilitan mejoras en la calidad y productividad del diseño y desarrollo. Ayudan a automatizar metodologías de software y desarrollo de aplicaciones.

### **2.4.1. Visual Paradigm**

Visual Paradigm (VP) para UML es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, la realización de ingeniería inversa<sup>1</sup> y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Las principales características que presenta el VP es que soporta UML versión 2.1, genera Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento. Permite un modelado colaborativo con CVS y Subversion, Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XML. Proporciona además una ingeniería inversa - Código a modelo, código a diagrama, Ingeniería inversa Java, C++, Esquemas XML, XML.NET exe/dll, CORBA IDL y la generación de código - Modelo a código, diagrama a código.

También posee un Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso. Diagramas EJB - Visualización de sistemas EJB. Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones. Diagramas de flujo de datos.

Otras de las características que presenta el VP son:

- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.

---

<sup>1</sup> Ingeniería Inversa: proceso ingenieril en el que se obtienen modelos conceptuales a partir de los artefactos de software como código fuente, ejecutables, binarios y ficheros intermedios.

- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

También incluye otras herramientas y plugins de modelado UML como es la Plataforma Java (Windows/Linux/Mac OS X): SDE para Eclipse, SDE para NetBeans, SDE para Sun ONE, SDE para Oracle JDeveloper, SDE para JBuilder, SDE para IntelliJ IDEA, SDE para WebLogic Workshop, Y para la Plataforma Windows: SDE para Microsoft Visual Studio. (Oficial, Bibliografía)

Se propone el uso de Visual Paradigm UML como herramienta CASE porque tiene UML 2.1 habilitado y este es un estándar ampliamente utilizado para el modelado del software. VP para UML es un producto galardonado que le facilita a las organizaciones la diagramación visual y el diseño de sus proyectos. Proporciona código y es compatible con hasta 10 lenguajes de programación. Brinda una alta interoperabilidad, además de ser multiplataforma.

## **2.5. Lenguaje de Programación**

Los lenguajes de programación son el vehículo de comunicación entre el hombre y la computadora. Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un programa informático. (Ravi, y otros, 1986)

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet los usuarios formularon demandas y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la web, que permitieran interactuar con los usuarios

y utilizaran Sistemas de Bases de Datos. Los lenguajes de programación web se dividen en dos grupos: del lado cliente y del lado del servidor.

### **Lenguaje de Programación del lado del servidor**

La programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

Programar del lado servidor tiene como gran ventaja que cualquier cosa puede hacerse sin tener en cuenta el tipo de cliente, ya que la aplicación se ejecuta en el servidor que es un ambiente controlado, una vez ejecutada la aplicación, el resultado que se envía al cliente puede estar en un formato normalizado que cualquier cliente puede mostrar. La desventaja es que el servidor se sobrecarga de trabajo ya que además de servir páginas es responsable de ejecutar aplicaciones.

### **PHP Hypertext Pre-processor**

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor. Algunas de sus principales características son:

- Es un lenguaje libre.
- Soporta gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, entre otras.
- 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, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- Permite el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Multiplataforma (Windows, Unix, Linux, Mac, etc.).
- Perfecta integración del Apache-PHP-MySQL, Oracle, MS-SQL, PostgreSQL.
- Bastante sencillo de aprender y utilizar.
- Interactúa con muchas librerías como: manejo de archivos PDF, Flash, calendarios, XML, IMAP, POP, gráficos en general. (Hinostroza, 2007)

Se utiliza este lenguaje de programación en su versión 5.3 porque es multiplataforma. Se caracteriza por ser un lenguaje muy rápido y fácil de aprender. Completamente orientado a la Web. Muestra una perfecta integración con los gestores de Base de Datos más utilizados y reconocidos internacionalmente. Capacidad de expandir su potencial utilizando módulos. Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones. No requiere definición de tipos de variables ni manejo detallado del bajo nivel. Además se selecciona porque es el lenguaje en el cual se encuentra implementado el framework<sup>2</sup> que se utiliza para la solución.

### **Lenguaje de Programación del lado del Cliente**

La programación del lado del cliente tiene como principal ventaja que la interpretación de la respuesta se delega al cliente, con lo cual se evita recargar al servidor de trabajo. El servidor sólo envía el código, y es tarea del navegador interpretarlo. La gran desventaja de esta opción de programación es que el código que el servidor envía es "sensible" a qué acciones puede o no hacer el navegador; pero gracias al empeño de desarrolladores esta desventaja que presentan los lenguajes de programación del lado del cliente ha sido minimizada con la aparición del protocolo https que se utiliza para encriptar la información que se envía y se recibe. Como otra desventaja que presentan es que el mismo código no va ser muy bien interpretado por algunos navegadores usados ya que ellos utilizarían diversas tecnologías por el lado del cliente.

### **HTML HyperTextMarkupLanguage**

HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. También esto constituye una ventaja con los nuevos paradigmas de la programación web orientadas a la web 2.0 que va orientada mayormente al trabajo con AJAX.

---

<sup>2</sup>Estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

### JavaScript

JavaScript es un lenguaje que está basado en objetos. No es, como Java, un lenguaje de programación orientado a objetos (OOP). JavaScript emplea clases y herencia, típicas de la OOP lo que no de la misma forma que otros lenguajes de programación. No es necesario declarar los tipos de variables que van a utilizarse. Las referencias a objetos se comprueban en tiempo de ejecución, por lo tanto no se compila. La ventaja que presenta JavaScript sobre el HTML es que permite crear páginas más dinámicas, lo que las hace más atractivas para el usuario. Es condición indispensable saber HTML para utilizar y dominar el JavaScript.

Con JavaScript se pueden utilizar varias librerías para facilitar el trabajo de los desarrolladores, tanto para el diseño como para crear funcionalidades. Para obtener un diseño más dinámico y agradable para el usuario se propone el uso de la librería ExtJS de JavaScript, la cual se define como:

Una librería JavaScript que permite construir aplicaciones web complejas. La librería ExtJS incluye: componentes de interfaz de usuario de alto nivel y personalizables. Modelo de componentes extensibles. Un API fácil de usar. Licencias Open source y comerciales.

Se decide el uso de esta librería para el diseño de la aplicación en su versión 3.3 porque permite crear aplicaciones complejas utilizando componentes predefinidos. Provee una experiencia consistente sobre cualquier navegador, evitando el problema de validar que el código escrito funcione bien en cada uno. Además la ventana flotante que provee ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido. Cuenta con un motor de render que permite crear un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo. El tráfico de red puede disminuir al permitir que la aplicación elija qué información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

## 2.6. Entorno de Desarrollo Integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como

C++, Python, Java, C#, Delphi, Visual Basic, etc. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, al que mediante plugins se le puede añadir soporte de lenguajes adicionales.

### **2.6.1. IDE NetBeans**

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE. Es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desplegados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

Se opta por el uso de este IDE en su versión 6.9.1 porque es libre y de código abierto. Tiene una interfaz muy amigable e intuitiva, característica fundamental en un IDE. Aunque consume recursos adicionales con respecto a otros IDEs y es uno de los más utilizados en cuanto a desarrollo de aplicaciones Web. Presenta un editor de formularios muy potente con respecto a otros y muestra buena operatividad con Symfony. Dada su buena integración con este Framework y que sea muy bueno para el desarrollo sobre la web es que se llega la conclusión de que es el que mejor se ajusta a las necesidades del proyecto.

## **2.7. Framework**

En el desarrollo de software, un Framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo que extiende o utiliza las aplicaciones del dominio.

### 2.7.1. *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. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. (Potencier & Zaninotto, 2008)

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix 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 solo 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 adaptable 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.

## 2.8. Servidor Web

Los servidores proporcionan un servicio al cliente y devuelven los resultados. La plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PCs poderosas, estaciones de trabajo, minicomputadores o sistemas grandes. Además deben manejar servicios

como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación y contabilidad.

Ejemplo de aplicaciones servidoras:

- Apache Web Server.
- FTP Server-U.
- Samba Server.
- Microsoft Exchange Server.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones o protocolo de comunicación, la cual proporciona los mecanismos básicos de direccionamiento y transporte.

Se denomina protocolo al conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o autómatas de otro tipo, tales como dispositivos electrónicos capaces de interactuar en una red.

### **2.8.1. Apache**

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Apache HTTP es de tecnología Open Source sólido y para uso comercial desarrollado por la Apache Software Foundation. Se propone el uso de este servidor web en su versión 2.0 porque es personalizable, la arquitectura modular de Apache permite construir un servidor hecho a la medida.

En cuanto a la administración los archivos de configuración de Apache están en ASCII, por lo que tiene un formato simple, y pueden ser editados tan solo con un editor de texto. Estos son transferibles, lo que permite la clonación efectiva de un servidor. El servidor puede ser administrado vía línea de comandos, lo que hace la administración remota muy conveniente.

Por otra parte se trata de un servidor muy eficiente. Mucho esfuerzo se ha puesto en optimizar el rendimiento del código "C" de Apache. Como resultado, este se ejecuta rápido y consume menos recursos de sistema en comparación a otros servidores. Además, Apache corre en una amplia variedad de sistemas operativos, incluyendo varias versiones de UNIX, Windows9x/NT, MacOS (Sobre Power PC), y varios otros.

El soporte de Apache es provisto por “The Apache Group” o “La Fundación Apache”, una gran cantidad de usuarios muy dedicados a su comunidad, así como compañías que ofrecen versiones pagadas de Apache.

## 2.9. Sistema Gestor de Bases de Datos (SGBD)

Los Sistemas de Gestión de Bases de Datos (en inglés Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Un sistema de base de datos es básicamente un sistema computarizado para guardar registros, cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones. La información en cuestión puede ser cualquier cosa que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración. (Date 2001)

### 2.9.1. SGBD. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Se opta por el uso de PostgreSQL como SGBD en su versión 8.4 porque se caracteriza por ser un sistema estable, de alto rendimiento, gran flexibilidad ya que puede funcionar sobre la mayoría de los sistemas Unix. Además, tiene características que permiten extender fácilmente el sistema. PostgreSQL puede ser integrado al ambiente Windows permitiendo de esta manera a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes. Permite desarrollar o migrar aplicaciones desde Access, Visual Basic, Foxpro, Visual Foxpro, C/C++ Visual C/C++, Delphi, etc., para que utilicen a PostgreSQL como servidor de BD.

## 2.10. Base de Datos Espacial

Base de datos espacial (spatial database) es un sistema administrador de bases de datos que maneja datos existentes en un espacio o datos espaciales. En este tipo de bases de datos es imprescindible establecer un cuadro de referencia (un SRE, Sistema de Referencia Espacial) para definir la localización y relación entre objetos, ya que los datos tratados en este tipo de bases de datos tienen un valor relativo, no es un valor absoluto. Los sistemas de referencia espacial pueden ser de dos tipos: georreferenciados (aquellos que se establecen sobre la superficie terrestre. Son los que normalmente se utilizan, ya que es un dominio manipulable, perceptible y que sirve de referencia) y no georreferenciados (son sistemas que tienen valor físico, pero que pueden ser útiles en determinadas situaciones).

## 2.11. PostGIS

PostGIS es un módulo que añade soporte de objetos geográficos al gestor objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la licencia pública general de GNU.

PostGIS ha sido desarrollado por la empresa canadiense Refraction Research, especializada en productos "Open Source". PostGIS es hoy en día un producto veterano que ha demostrado versión a versión su eficiencia. En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL, y a juicio de muchos, es muy similar a la versión geográfica de la archiconocida Oracle.

Un aspecto que hay que tener en cuenta es que PostGIS ha sido certificado en el 2006 por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas también interoperables. PostGIS almacena la información geográfica en una columna del tipo GEOMETRY.

Se utiliza esta base de datos espacial porque presenta una gestión de datos centralizada. Fortaleza fundamental para el equipo de desarrollo que necesita trabajar con los datos organizados de esa forma. Permite una edición multiusuario. Admite el almacenamiento de los datos en una SGBD como PostgreSQL. Además de permitir el acceso remoto a datos y aplicaciones externas.

## 2.12. Conclusiones Parciales

Tras analizar las distintas tendencias y tecnologías propuestas en el presente capítulo se escogieron cuál de las disímiles metodologías de desarrollo de software así como de la variedad de herramientas CASE existentes, habrían de utilizarse en la elaboración de la documentación ingenieril correspondiente a los subsistemas que se implementan. También se estudiaron las herramientas que ayudarán a la construcción de la solución.

Luego de culminar con el enfoque general de lo que ha tratado el capítulo, se llega a la conclusión de que la metodología a utilizar por los desarrolladores en las actividades de construcción del software será el RUP, el Visual Paradigm será la herramienta Case con la que se realizará el modelado de los procesos y artefactos arrojados por la metodología escogida y UML 1.0 como lenguaje de modelado. La solución se desarrolla en un ámbito web por lo que el lenguaje de programación a utilizar del lado del servidor será PHP y del lado cliente serán utilizados HTML y JavaScript con su librería Ext. Por las ventajas que brinda el uso de un framework en la implementación de soluciones para la web es que se selecciona Symfony para la construcción de los subsistemas; además se decide que sea PostgreSQL el gestor de base de datos.

## CAPÍTULO 3. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

### 3.1. Introducción

En el presente capítulo se enumeran y especifican los Requisitos Funcionales (RF) y los Requisitos No Funcionales (RNF) que debe cumplir el sistema, se identifican los actores y casos de uso del sistema para conformar el Diagrama de Casos de Uso del Sistema (DCUS). Unido a esto se realiza una descripción detallada de los casos de uso que forman parte de la solución. También se estructuran los Diagramas de Clases del Diseño correspondientes a los casos de uso identificados con anterioridad. Finalmente se estructura el Modelo de Despliegue de la aplicación.

### 3.2. Modelo de Dominio en el Desarrollo de Software

El Modelo de Dominio es empleado fundamentalmente cuando los flujos de información son difusos, es decir, que tengan múltiples orígenes y cuando son sólo eventos o sucesos. También la imposibilidad de realizar subsistemas en el proceso de desarrollo del software dado por las múltiples interconexiones, es uno de los factores que influyen en la decisión de realizar un modelo de este tipo. Influye además en esta decisión, la existencia del solapamiento de responsabilidades y la dificultad en el establecimiento de las reglas del funcionamiento del producto a implementar.

Un Modelo del Dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, además de que no incluyen las responsabilidades de las personas que ejecutan las actividades. Según RUP este tipo de modelo representa un subconjunto del Modelo de Objeto del Negocio, lo cual no significa que siempre que haya un Modelo de Dominio tenga que existir obligatoriamente un Modelo de Negocio. Por lo que el Modelo de Dominio no es más que la representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés.

#### 3.2.1. *Modelo de Dominio de SIG-INRH*

Teniendo en cuenta que no se tienen bien definidos los procesos del negocio, ni los responsables de dichos procesos, se realizará una modelación del dominio y se procederá a explicar cada uno de los conceptos que forman parte del mismo. Todo ello para tener una mejor comprensión de la estructura y dinámica de la organización.

Se muestra el modelo de dominio correspondiente al entorno donde estará el sistema teniendo en cuenta el proceso de representación de la información geográfica, conceptos, actividades, personas involucradas en ello y las reglas del negocio.

**3.2.2. Diagrama de clases del modelo del dominio**

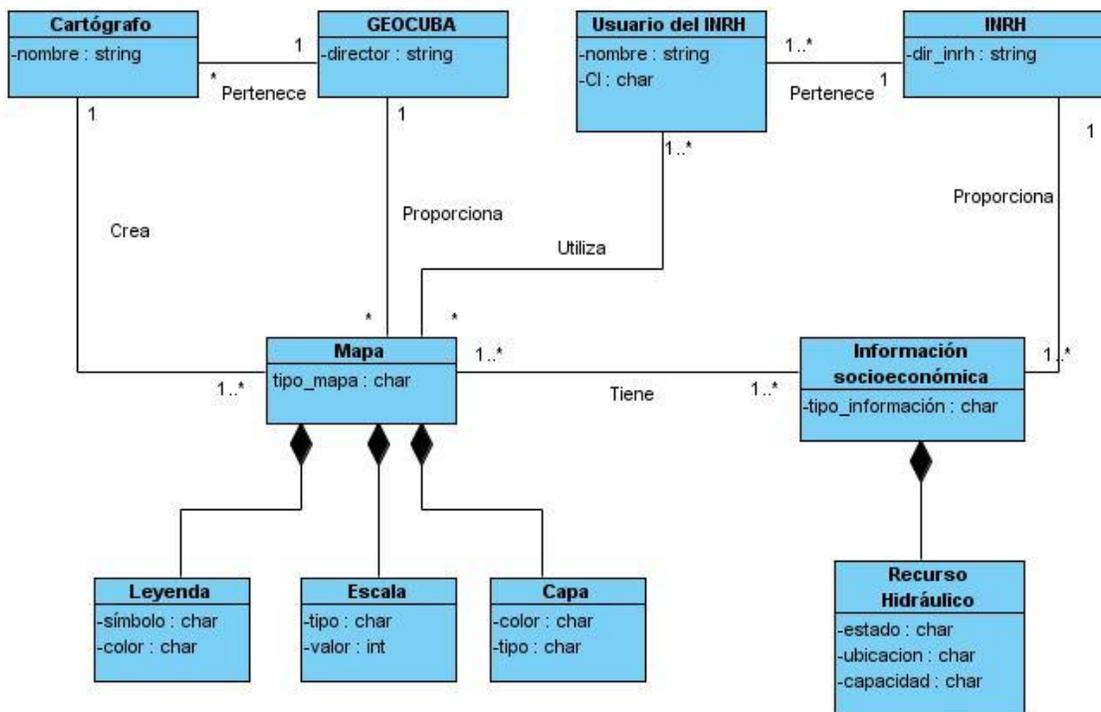


Ilustración 1 Diagrama de Clases del Dominio.

**Definición de las clases del modelo del dominio**

**Cartógrafo**

Persona que se dedica profesionalmente a la realización de cartas geográficas, al estudio y elaboración de mapas.

**GeoCuba**

GEOCUBA es un grupo empresarial que se dedica a la elaboración, producción y venta de planos, mapas y cartas náuticas con diversos fines, así como a la realización de estudios geográficos, de impacto ambiental, e investigaciones científicas en ramas del campo de las geociencias, entregando a sus clientes, productos informativos terminados con una alta calidad y fiabilidad.

### **Usuario INRH**

Persona que interactúa con el sistema del cual necesite extraer algún tipo de información.

### **INRH**

Es la institución que se encarga de la administración y control de todos los objetivos hidráulicos cubanos.

### **Mapa**

Es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráqueos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancia, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto.

### **Escala**

Relación entre la distancia que separa dos puntos en un mapa y la distancia real de esos dos puntos en la superficie terrestre. En los mapas, la escala puede expresarse de tres modos distintos: en forma de proporción o fracción, con una escala gráfica o con una expresión en palabras y cifras. Cuanto mayor es la escala, más se aproxima al tamaño real de los elementos de la superficie terrestre. Los mapas a pequeña escala generalmente representan grandes porciones de la Tierra y, por tanto, son menos detallados que los mapas realizados con escalas más grandes.

La relación matemática entre las dimensiones en el mapa, carta o plano y la superficie terrestre que representa. Por extensión puede referirse a la mayor o menor profundidad del enfoque en un tema geográfico.

### **Leyenda**

Explicación de los símbolos, los colores, las tramas y los sombreados empleados en un mapa; suele encontrarse a pie de página o en un recuadro, situado en sus márgenes o bien en su dorso. Los símbolos empleados en los mapas pueden llegar a contener un gran volumen de información, que por su facilidad de lectura permiten una rápida interpretación.

### **Recurso Hidráulico**

Comprende los diferentes tipos de objetivos hidráulicos que existen sobre el territorio nacional, se refiere a presa, ríos y embalses.

### **Información Socioeconómica**

Es un conjunto organizado de datos procesados referentes al aspecto social y económico de cualquier lugar de interés del país.

### **Breve descripción del diagrama**

GeoCuba es la empresa que determina todos los procesos geológicos desarrollados en Cuba, a la que pertenecen varios cartógrafos que son los encargados de crear o construir los mapas, que se le proporcionan a otras entidades.

Los mapas están compuestos por varias escalas representativas, leyendas que permiten un mejor entendimiento de los mismos y la tipografía, que está referida a la variedad de mapas existentes de acuerdo con su especificación, que pueden ser de División Político-Administrativa, Vegetación, Yacimientos Geológicos, entre otros. También estos mapas tienen en su haber toda la información socioeconómica, tratada en los distintos tipos de datos referentes a todos los sectores de la economía y la sociedad. Los mapas son utilizados además, por los clientes pertenecientes al INRH, y este último es el responsable de brindar toda la información socioeconómica que se le agregará posteriormente a dichos mapas.

## **3.3. Requisitos**

El proceso de construcción de un software está dado por disímiles actividades que han de ser monitoreadas por alguna guía metodológica, garantizando así la calidad del producto y la satisfacción de los clientes y desarrolladores al mismo tiempo. Durante la construcción del software se llevan a cabo un grupo de técnicas y procedimientos de acuerdo con la metodología seleccionada para la construcción de la solución, arrojando así la documentación relacionada con cada uno de los flujos de trabajo.

Una etapa inicial y muy importante dentro del proceso de la Ingeniería de Software, es la Ingeniería de Requisitos, donde se realiza el proceso de descubrir, analizar, escribir y verificar los servicios y restricciones, del sistema de software que se desea producir; este proceso se realiza mediante la obtención, el análisis, la especificación, la validación y la administración de los requisitos del software. Los requisitos de software son

las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar (Martínez, 2004).

Para la obtención de los requisitos de los subsistemas se hizo necesario acudir a varias técnicas relacionadas con dicho proceso como es la entrevista a varios especialistas del departamento de Geoinformática del centro GEYSED, departamento que se encarga del desarrollo de SIG para distintas empresas que así lo quieran. También se entrevistó a especialistas del centro UCID que desarrollan aplicaciones para realizar la administración desde un entorno web. Unido a esto, se utilizó la técnica llamada Lluvia de Ideas así como la del Enfoque Grupal, donde cada uno de los integrantes del equipo de desarrollo expresa su punto de vista e ideas relacionadas con las funcionalidades que requiera el producto. Además, se contó con el previo estudio e investigación de las herramientas y tecnologías de software en los se desarrollan sistemas similares.

Al lograr identificar los requisitos iniciales del software y realizar el análisis de los mismos mediante la técnica de verificación, donde se limaron las ambigüedades y la falta de consistencia, se arrojaron los requisitos funcionales y no funcionales que se presentan en los subepígrafes que se relacionan a continuación.

### **3.3.1. Requisitos Funcionales**

Son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales no alteran la funcionalidad del producto, esto quiere decir que los requisitos funcionales se mantienen invariables sin importarle con qué propiedades o cualidades se relacionen. (Jacobson, y otros, 2000)

**RF1.** El sistema debe permitirle al administrador adicionar usuarios al sistema.

**RF2.** El sistema debe permitirle al administrador modificar los usuarios al sistema.que se encuentran añadidos al sistema.

**RF3.** El sistema debe permitirle al administrador eliminar los usuarios al sistema.que se encuentran añadidos al sistema.

**RF4.** El sistema debe permitir mostrar un listado con todos los usuarios que se encuentran añadidos al sistema.

**RF5.** El sistema debe permitirle al administrador asignarle roles a los usuarios que se encuentran añadidos al sistema.

**RF6.** El sistema debe permitirle al administrador cambiar la contraseña de los usuarios que se encuentran añadidos al sistema.

**RF7.** El sistema debe permitirle al administrador asignar usuarios a los sistemas que se quiera tenga permisos.

**RF8.** El sistema debe permitirle al administrador adicionar roles al sistema.

**RF9.** El sistema debe permitirle al administrador modificar los roles que se encuentran añadidos en el sistema.

**RF10.** El sistema debe permitirle al administrador eliminar los roles que se encuentran añadidos en el sistema.

**RF11.** El sistema debe permitir mostrar un listado con todos los roles que se encuentran añadidos al sistema.

**RF12.** El sistema debe permitirle al administrador asignarle acciones a los roles que se encuentran añadidos al sistema.

**RF13.** El sistema debe permitirle al administrador adicionar sistemas.

**RF14.** El sistema debe permitirle al administrador modificar los sistemas que se encuentran añadidos.

**RF15.** El sistema debe permitirle al administrador eliminar los sistemas que se encuentran añadidos.

**RF16.** El sistema debe permitir mostrar un listado con todos los sistemas que se encuentran añadidos.

**RF17.** El sistema debe permitirle al administrador adicionar acciones al sistema.

**RF18.** El sistema debe permitirle al administrador modificar las acciones que se encuentran añadidas al sistema.

**RF19.** El sistema debe permitirle al administrador eliminar las acciones que se encuentran añadidas al sistema.

**RF20.** El sistema debe permitir mostrar un listado con todas las acciones que se encuentran añadidas al sistema.

**RF21.** El sistema debe permitirle al administrador adicionar funcionalidades al sistema.

**RF22.** El sistema debe permitirle al administrador modificar las funcionalidades que se encuentran añadidas al sistema.

**RF23.** El sistema debe permitirle al administrador eliminar las funcionalidades que se encuentran añadidas al sistema.

**RF24.** El sistema debe permitir mostrar un listado con todas las funcionalidades que se encuentran añadidas al sistema.

**RF25.** El sistema debe permitir que un usuario que se encuentre registrado pueda iniciar sesión e interactuar con las funcionalidades que le corresponden.

**RF26.** El sistema debe permitirle al administrador buscar un recurso hidráulico que se encuentre en el sistema.

**RF27.** El sistema debe permitirle al administrador modificar los campos que sean cambiables de información socioeconómica correspondiente a los recursos hidráulicos que se encuentran en el sistema.

**RF28.** El sistema debe permitir mostrar un listado con todos los recursos hidráulicos que se encuentran en el sistema.

### ***Requisitos no Funcionales***

Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. (Jacobson, y otros, 2000)

#### **RNF 1 Usabilidad**

El sistema podrá ser usado por personas con conocimientos básicos en el manejo de computadoras. Cuenta con una interfaz amigable e intuitiva que sirve de ayuda al usuario para una mejor interacción con las funcionalidades que brinda pues cada una está identificada con imágenes que se relacionan con la funcionalidad que debe desarrollar.

#### **RNF 2 Fiabilidad**

Para que el usuario pueda realizar funcionalidades avanzadas en el sistema debe estar autenticado y de acuerdo con el rol que le sea asignado tendrá los permisos asociados.

#### **RNF 3 Requisitos del Software**

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

Para los servidores:

- Sistemas operativos GNU/Linux o Windows Server 2000 o superior.
- Servidor Web Apache 2.0 o superior, con módulo PHP 5 configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- PostGIS como extensión de PostgreSQL como soporte de datos espaciales.

Para las PCs clientes:

- Un Navegador como Mozilla Firefox, Chrome, Safari u otro navegador que cumpla con los estándares W3C.
- Sistema operativo: GNU/Linux, Windows y Mac OS (Macintosh).

#### **RNF 4 Requisitos del Hardware**

Para las PCs clientes:

- Se requiere tengan tarjeta de red.
- Al menos 256 MB de memoria RAM.
- Procesador 512 MHz como mínimo.

Para los servidores:

- Se requiere tarjeta de red.
- Al menos 2GB de RAM y 10GB de disco duro.
- Procesador 3 GHz como mínimo.

#### **RNF 5 Confiabilidad**

La información manejada por el sistema está protegida de accesos no autorizados y divulgación. Para una mejor seguridad se hicieron dos modelos de datos para separar de la base de datos que contenga la información asociada al sistema la que contenga la referida a los usuarios y contraseñas. Como se trata de un sistema de gestión requiere mantenimiento y optimización en el almacenamiento, se estima un tiempo

promedio de 6 meses entre posibles fallas y el tiempo medio de reparación, en caso de un fallo es de diez horas.

#### **RNF 6 Rendimiento**

El tiempo de respuesta oscila entre 0.22 segundos y 2 segundos, pero puede variar de acuerdo a la cantidad de usuarios que estén solicitando autenticación y por la cantidad de información que se necesite procesar ante una petición del usuario.

Al igual que el tiempo de respuesta, la velocidad de procesamiento de la información, la actualización y la recuperación dependerán de la cantidad de información que tenga que procesar la aplicación.

#### **RNF 7 Restricciones de diseño**

Diseño sencillo, con pocas entradas, donde no sea necesario mucho entrenamiento para utilizar el sistema. El producto de software final se diseñará sobre una arquitectura cliente-servidor. Se deben emplear los estándares establecidos (diseño de interfaces, base de datos y codificación).

#### **RNF 8 Portabilidad**

El sistema será multiplataforma y su puesta en marcha debe ser compatible con los sistemas operativos, Windows, Linux.

#### **RNF 9 Legales**

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

La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

Como producto, se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

#### **RNF 10 Seguridad**

Los usuarios deben autenticarse antes de entrar al sistema. Garantizar el acceso controlado a la información, se mostrarán las interfaces a los usuarios, en dependencia de su nivel de acceso.

### 3.4. Descripción del sistema propuesto. Modelo de Casos de Uso del Sistema

Luego de haber realizado la descripción de los requisitos, los cuales constituyen las funcionalidades esenciales del sistema, se le da paso a la siguiente etapa de conceptualización de un software, que es la descripción del Modelo de Casos de Uso del Sistema (DCUS), que contiene actores, casos de uso y sus respectivas relaciones.

Los requisitos del software se agrupan según sus características, utilidad y usabilidad conformando así los llamados casos de uso, los cuales no son más que fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

Estos casos de uso describen cómo se comporta y funciona un sistema específico, mediante un documento que narra de manera secuencial las acciones relevantes que realiza un actor en interacción con el sistema hasta completar un proceso, sin darle importancia a los detalles de la implementación.

Los actores se definen como los roles que puede tener un usuario, pueden ser humanos, otros sistemas, máquinas, hardware, etc. que interactúan con un sistema para de esta forma intercambiar datos, aunque en algunos casos pueden constituir un recipiente pasivo de información. (MOLINA and IBÁÑEZ 2001).

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. SIG-INRH cuenta con tres actores los cuales se especifican a continuación.

Actor	Descripción
Usuario	Son los especialistas o administradores que necesitan autenticarse en la aplicación.
Especialista	Es la persona miembro del INRH que tendrá permisos para modificar la información que se muestra.

Administrador	Es el encargado de controlar toda la información referente a los usuarios y de administrar el sistema.
---------------	--

Tabla 1 Descripción de los actores del sistema.

### 3.4.1. Diagrama de casos de Uso del Sistema

La solución al problema planteado cuenta con un DCUS, el cual se presenta a continuación. Este diagrama representa gráficamente a los procesos y su interacción con los actores.

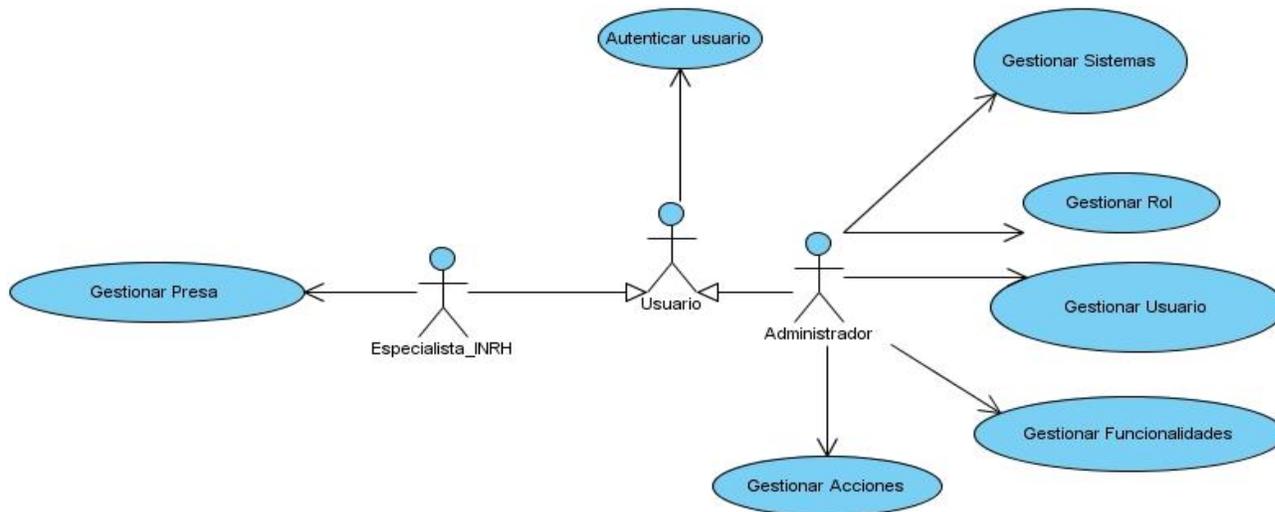


Ilustración 2 Diagrama de casos de uso del sistema.

### 3.4.2. Descripción Textual de Casos de Uso del Sistema

Debido a que las descripciones textuales son muy extensas solo se muestra la descripción del caso de uso Gestionar Usuario, el resto de las descripciones textuales se encuentran en el Anexo #1.

- CU Gestionar Usuarios

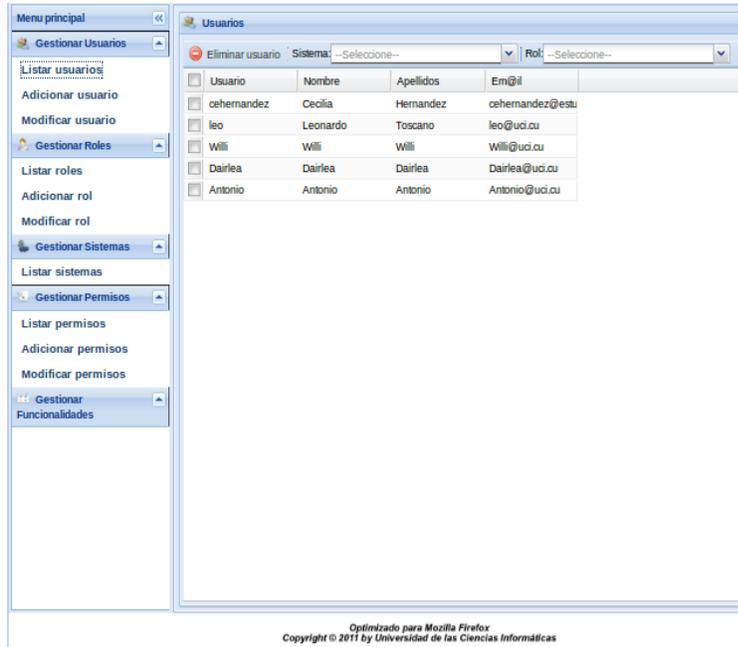
<b>Caso de Uso:</b>	Gestionar Usuarios
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar, eliminar o insertar un usuario al sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario administrador desea insertar, modificar, asignar sistemas, roles, cambiar clave o eliminar un usuario y termina cuando haya sido ejecutada alguna de las acciones anteriormente especificadas..
<b>Precondiciones:</b>	El usuario tiene que estar autenticado.
<b>Referencias</b>	RF 1, RF 2, RF 3, RF 4, RF 5, RF 6, RF 7,
<b>Prioridad</b>	Crítico

**Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el usuario desea realizar una de las opciones que se especifican a continuación: (Ver Interfaz 1)</p> <ul style="list-style-type: none"> <li>- Listar Usuarios.</li> <li>- Adicionar Usuario.</li> <li>- Modificar Usuario.</li> <li>- Asignar Roles.</li> <li>- Asignar Sistema.</li> <li>- Cambiar Contraseña.</li> <li>- Eliminar Usuario.</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Listar Usuarios”, ver sección “Listar Usuarios”.</li> <li>- Si selecciona “Adicionar Usuario”, ver sección “Adicionar Usuario”.</li> <li>- Si selecciona “Modificar Usuario”, ver sección “Modificar Usuario”.</li> <li>- Si selecciona “Asignar Roles”, ver sección “Asignar Roles”.</li> <li>- Si selecciona “Asignar Sistema”, ver sección “Asignar Sistema”.</li> <li>- Si selecciona “Cambiar Contraseña”, ver</li> </ul>

	<p>sección “Cambiar Contraseña”.</p> <p>- Si selecciona “Eliminar Usuario”, ver sección “Eliminar Usuario”.</p>
	<p><b>3.</b> El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario.</p>
<p><b>Prototipo de Interfaz</b></p>	
<p><b>Interfaz 1</b></p> 	
<p><b>Sección “Listar Usuarios”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p><b>3.</b> El usuario selecciona la opción Listar Usuario (Ver Interfaz 1)</p>	<p><b>4.</b> El sistema comprueba que existan usuarios registrados y procede a mostrar una tabla que contiene el listado de todos los usuarios que se encuentran añadidos al sistema. (Ver Interfaz 2)</p>

Interfaz 2



Flujos Alternos

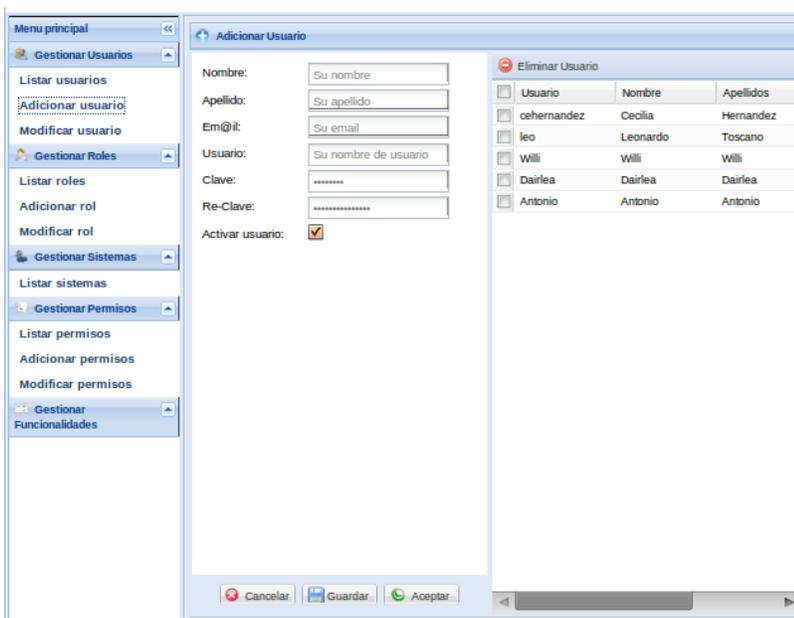
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje de error indicando que no hay usuarios añadidos al sistema finalizando así el caso de uso.

Sección "Adicionar Usuario"

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Adicionar Usuario (Ver Interfaz 1)	4. El sistema muestra un formulario que le permite al usuario insertar los datos (Nombre, Apellido, Email, Usuario y Contraseña).
5. El usuario inserta los datos (Nombre, Apellido, Email, Usuario y Contraseña) y da clic en el botón guardar.	6. El sistema valida que todos los campos estén correctos.

	7. El sistema verifica que el usuario no se encuentre insertado.
	8. El sistema guarda el nuevo usuario manteniendo la interfaz de Adicionar Usuario finalizando así el caso de uso.

**Interfaz 3**



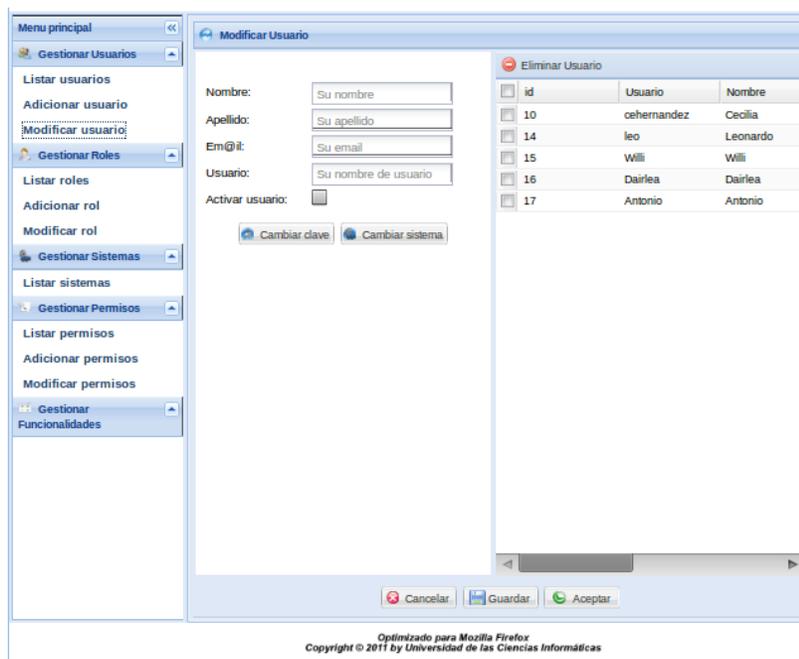
Optimizado para Mozilla Firefox  
Copyright © 2011 by Universidad de las Ciencias Informáticas

**Flujo Alterno**

Acción del Actor	Respuesta del Sistema
5.1 El usuario inserta los datos (Nombre, Apellido, Email, Usuario y Contraseña) y da clic en el botón Aceptar.	8.1 Realiza los pasos 6 y 7 del flujo normal de los eventos, guarda el usuario y procede a cerrar la interfaz de Adicionar Usuario, finalizando así el caso de uso.
	6.1 El sistema le informa al usuario que hay error en la entrada de los datos y el usuario vuelve al paso 3 del flujo normal de los eventos.

	<p><b>7.1</b> El sistema envía un mensaje de error especificando que el usuario se encuentra registrado en el sistema, y vuelve al paso 3 si desea adicionar otro usuario, en caso contrario finaliza el caso de uso.</p>
<p><b>Sección “Modificar Usuario”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p><b>3.</b> El usuario selecciona la opción Modificar Usuario (Ver Interfaz 1)</p>	<p><b>4.</b> Ir al paso 2 del flujo normal de los eventos de la sección Listar Usuarios.</p>
<p><b>5.</b> El usuario selecciona el usuario que desea modificar. (Ver Interfaz 4)</p>	<p><b>6.</b> El sistema llena el formulario con los datos que tiene guardado del usuario seleccionado.</p>
<p><b>7.</b> El usuario modifica los datos (Nombre, Apellido, Email, Usuario y Contraseña) y da clic en el botón guardar.</p>	<p><b>8.</b> El sistema valida que todos los campos estén correctos.</p>
	<p><b>9.</b> El sistema guarda los cambios realizados por el usuario, manteniendo la interfaz de Modificar Usuario, finalizando así el caso de uso.</p>

Interfaz 4



Flujo Alterno

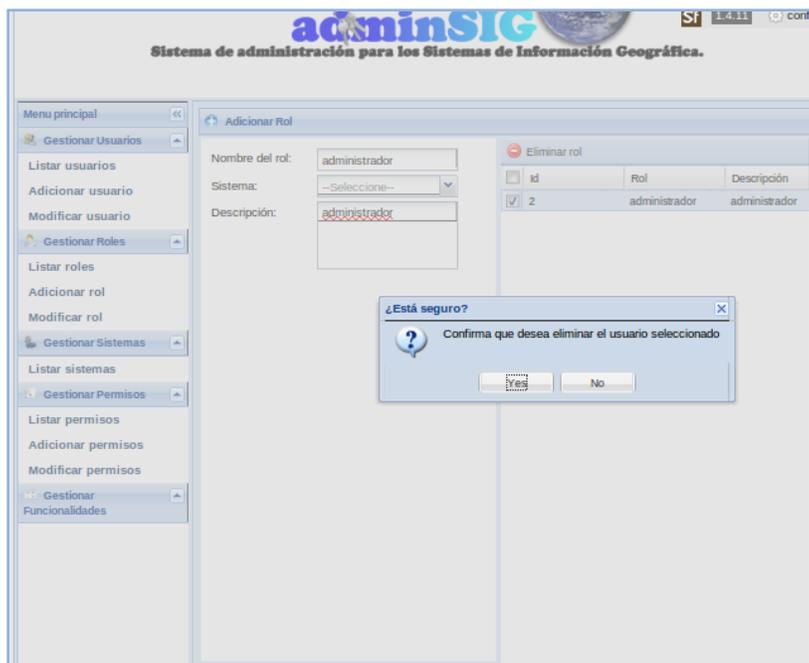
Acción del Actor	Respuesta del Sistema
7.1. El usuario modifica los datos (Nombre, Apellido, Email, Usuario y Contraseña) y da clic en el botón Aceptar.	9.1. El sistema realiza el paso 8 del flujo normal de eventos y procede a cerrar la interfaz de Modificar Usuario, finalizando así el caso de uso.
	8.1. El sistema muestra un mensaje indicando que hay error en la entrada de los datos y vuelve al paso 4.

Sección “Eliminar Usuario”

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

3. El usuario selecciona alguna de las opciones disponibles desde la interfaz 1. (Modificar Usuario o Listar Usuarios)	4. El sistema ejecuta el paso 4 del flujo normal de los eventos de las secciones Modificar Usuario o Listar Usuarios
5. El usuario selecciona al usuario que desea eliminar y da clic en la opción Eliminar Usuarios.	6. El sistema muestra un mensaje al usuario pidiéndole confirmación de la acción que desea realizar.
7. El usuario confirma la acción.	8. El sistema elimina al usuario seleccionado por el usuario y finaliza el caso de uso.

**Interfaz 5**

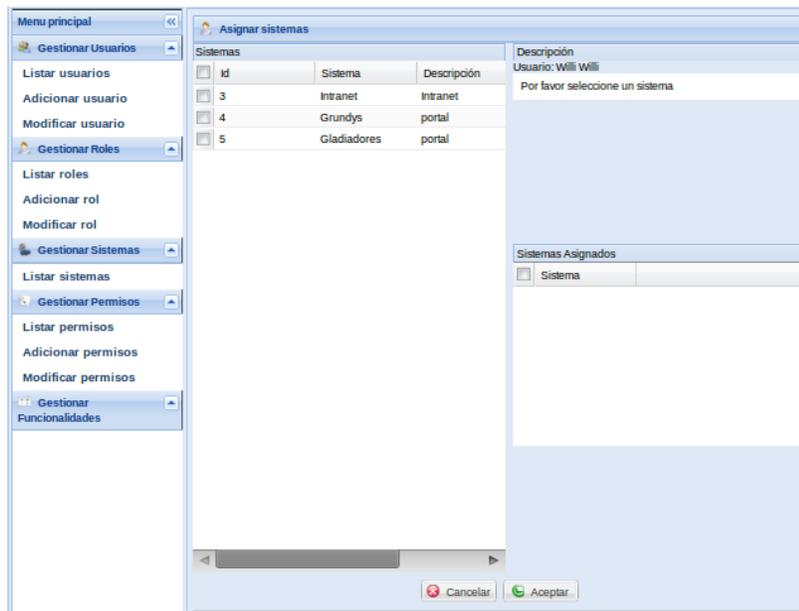


**Flujo Alterno.**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5.1. El usuario no selecciona ningún usuario y da clic en la opción Eliminar Usuarios.	6.1 El sistema muestra un mensaje donde se especifica que debe seleccionar un usuario.
7.1 El usuario no confirma la acción.	8.1. El sistema no elimina al usuario

	seleccionado por el usuario y finaliza el caso de uso.
<b>Sección “Asignar Sistemas”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El usuario selecciona la opción Modificar Usuario.	4. El sistema muestra la interfaz Modificar Usuario.
5. El usuario selecciona al usuario que le desea añadirle los sistemas y da clic en la opción Cambiar Sistemas.	6. El sistema muestra la interfaz Asignar sistemas.
7. El usuario selecciona un sistema de la lista de sistemas que no le han sido asignados al usuario seleccionado y da doble clic.	8. El sistema guarda el sistema seleccionado en la lista de sistemas añadidos al usuario.
9. El usuario da clic en el botón Aceptar.	10. El sistema cierra esta interfaz y vuelve a la interfaz de Modificar Usuario.

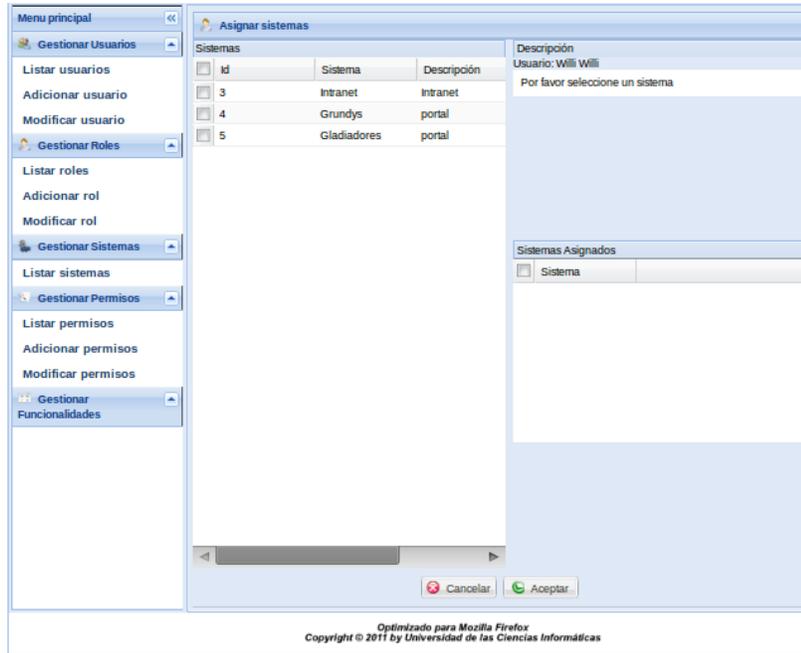
**Interfaz 6**



Optimizado para Mozilla Firefox  
Copyright © 2011 by Universidad de las Ciencias Informáticas

Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	6.1. El sistema muestra un mensaje indicando que debe seleccionar un usuario para poder asignarle sistemas
Sección "Asignar Roles"	
Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Modificar Usuario.	4. El sistema muestra la interfaz Modificar Usuario.
5. El usuario selecciona al usuario que le desea añadir los roles y da clic en la opción Cambiar Rol.	6. El sistema muestra la interfaz Asignar Roles.
7. El usuario selecciona un rol de la lista de roles que no le han sido asignados al usuario seleccionado y da doble clic.	8. El sistema guarda el rol seleccionado en la lista de roles añadidos al usuario.
9. El usuario da clic en el botón Aceptar.	10. El sistema cierra esta interfaz y vuelve a la interfaz de Modificar Usuario.

Interfaz 7



**Flujo Alterno**

Acción del Actor	Respuesta del Sistema
	6.1. El sistema muestra un mensaje indicando que debe seleccionar un usuario para poder asignarle roles.

**Sección “Cambiar contraseña”**

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Modificar Usuario.	4. El sistema muestra la interfaz Modificar Usuario.
5. El usuario selecciona al usuario que le desea cambiar la contraseña y da clic en la opción cambiar contraseña. (Ver Interfaz 8)	6. El sistema muestra la interfaz Cambiar Contraseña.

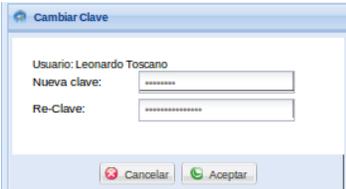
7. El usuario ingresa la nueva contraseña.	8. El sistema valida que el campo esté lleno y guarda los cambios.
<p><b>Interfaz 8</b></p> 	
<b>Flujo Alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	6.1 El sistema muestra un mensaje indicando que debe seleccionar un usuario para poder cambiarle la contraseña.
<b>Postcondiciones</b>	El sistema inserta, elimina, modifica, le asigna roles, sistemas y le cambia contraseñas a los usuarios a partir de la opción que se desee realizar.

Tabla 2 Descripción Textual CU Gestionar Usuarios.

### 3.5. Conclusiones Parciales

En el presente capítulo se han tratado de manera específica los artefactos generados en los dos primeros flujos de trabajo que definidos por RUP, lo cual representa la metodología escogida para llevar a cabo los procesos ingenieriles referentes al desarrollo de los subsistemas de administración y gestión de la información del SIGINRH.

La realización del modelo de dominio permitió identificar y detallar los principales eventos, términos y conceptos presentes en el entorno que se investiga garantizando así el establecimiento de las bases para identificar las funcionalidades del sistema que se desea construir. La realización del proceso de captura de requisitos concluyó con la obtención de las condiciones y las cualidades que el sistema debe cumplir para

poder funcionar correctamente y cumplir con las expectativas del cliente. Unido a esto se debe especificar que el levantamiento de los requisitos no funcionales, le dan al software el nivel de calidad que debe cumplir. También se muestra a través del análisis, un Modelo de Casos de Uso que permitió tener una vista global del sistema a desarrollar. Además, garantizó que a partir de las descripciones textuales de los casos de uso identificados se tenga la idea exacta de qué se quiere llevar a cabo con cada funcionalidad, a partir de la acción del actor y la respuesta que debe dar el sistema.

## CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### 4.1. Introducción

Para lograr una exitosa implementación de los subsistemas de administración y gestión de la información correspondientes al SIGINRH, es necesario realizar un estudio previo en cuanto a la estructura que ha de tener, diseñando de forma explícita las relaciones existentes entre cada una de las entidades que forman parte del análisis. En el capítulo que comienza se propone como parte de la solución a la problemática existente, el Modelo de Diseño que se corresponde con el flujo de trabajo de Análisis y Diseño propuesto por RUP. “El objetivo de la fase de diseño es la generación de una descripción sobre cómo sintetizar los objetos extraídos del dominio de la solución propuesta al problema” donde “el principal resultado del diseño es el modelo del diseño, que se esfuerza en conservar la estructura del sistema, impuesta por el modelo de análisis, y que sirve como esquema para la implementación.”(Lopez, 2003).

En el actual capítulo se ha de mostrar una vista general del sistema a construir, a partir del diagrama de paquetes organizado, así como algunos de los artefactos que forman parte del modelo de diseño en piezas manejables, además de realizar cada uno de los diagramas de clases del diseño correspondientes a cada uno de los casos de uso con que cuenta la aplicación, teniendo en cuenta la arquitectura y los patrones de diseño definidos para la misma.

### 4.2. Arquitectura de Software

Una definición reconocida de Arquitectura de Software(AS) es la de Clements: La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema.

Estas cuestiones estructurales se vinculan con el diseño, pues la AS es después de todo una forma de diseño de software que se manifiesta tempranamente en el proceso de creación de un sistema; pero este diseño ocurre a un nivel más abstracto que el de los algoritmos y las estructuras de datos. En el que muchos consideran un ensayo seminal de la disciplina, Mary Shaw y David Garlan sugieren que dichas cuestiones estructurales incluyen organización a grandes rasgos y estructura global de control; protocolos para la comunicación, la sincronización y el acceso a datos; la asignación de funcionalidad a elementos del diseño;

la distribución física; la composición de los elementos de diseño; escalabilidad y rendimiento; y selección entre alternativas de diseño.

Se puede concluir que el concepto más completo de AS es el que brinda la IEEE cuando define como AS a la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

IEEE 1471-2000

Una arquitectura define varios estilos arquitectónicos, entre los más conocidos se destacan:

- Arquitectura centrada en los datos.
- Arquitectura centrada en los flujos de datos.
- Arquitectura llamada y respuesta (call and return).
- Arquitectura Orientada a Objetos.
- Arquitectura en capas.

#### **4.2.1. Estilos Arquitectónicos**

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción.

Existen muchos estilos los cuales agrupan varios patrones arquitectónicos entre los que se destacan:

- Estilos de Flujo de Datos  
Tubería y filtros.
- Estilos Centrados en Datos  
Arquitecturas de pizarra o repositorio.
- Estilos de llamada y retorno  
Model-View-Controller (MVC).  
Arquitectura en capas.  
Arquitecturas orientadas a objetos.  
Arquitecturas basadas en componentes.
- Estilos de código móvil  
Arquitectura de máquinas virtuales.

- Estilos heterogéneos
  - Sistemas de control de procesos.
  - Arquitecturas basadas en atributos.
- Estilos Peer-to-Peer
  - Arquitecturas basadas en eventos.
  - Arquitecturas orientadas a servicios.
  - Arquitecturas basadas en recursos.

Después de analizados los estilos y vistos los patrones que engloban cada uno de ellos, se decidió el uso del estilo de llamada y retorno, y dentro del mismo el patrón Modelo Vista Controlador, que es el patrón arquitectónico que implementa Symfony como framework para la construcción de la solución.

### **Patrón Modelo Vista Controlador desarrollado en Symfony**

El *framework* seleccionado, al igual que la mayoría de los existentes para PHP está desarrollado bajo las restricciones, suposiciones y recomendaciones de la Arquitectura Modelo-Vista- Controlador (MVC).

MVC sugiere la separación del software en tres componentes: Modelo, Vista y controlador, los cuales serán explicados brevemente:

- Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí lo constituyen los datos puros y la lógica de los propios datos que puestos en el contexto del sistema proveen de información al usuario y en algunos casos a la propia aplicación.
- Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web, la “Vista” sería una página HTML con contenido dinámico sobre la cual el usuario puede realizar sus operaciones.
- Controlador: Es la parte encargada de manejar y responder las solicitudes del usuario, procesando toda la información necesaria y modificando el Modelo en caso de ser necesario.

#### *Ciclo de vida del MVC*

El ciclo de vida del MVC es normalmente representado por los tres componentes presentados anteriormente y el cliente (también conocido como usuario o actor). El primer paso en el ciclo de vida empieza cuando el usuario hace una solicitud al controlador con información sobre lo que él desea realizar. Posteriormente el controlador decide a quién debe delegar la tarea y es aquí donde el modelo empieza su trabajo. En esta

etapa, el modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador. Luego de terminada su labor, le regresa al controlador la información resultante de sus operaciones, el cual a su vez redirige a la vista la cual se encarga de transformar los datos en información visualmente entendible para el usuario. (Propuesta de diseño para proyectos informáticos que utilizan Symfony como Framework, 2010)

#### **4.2.2. Patrones de Diseño**

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Un patrón de diseño identifica: *Clases, Instancias, Roles, Colaboraciones* y la *distribución*.

*Un patrón de diseño es:* una solución estándar para un problema común de programación. Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. Un proyecto o estructura de implementación que logra una finalidad determinada. Un lenguaje de programación de alto nivel. Una manera más práctica de describir ciertos aspectos de la organización de un programa. Conexiones entre componentes de programas. La forma de un diagrama de objeto o de un modelo de objeto.

Los patrones de diseño se encuentran divididos en dos grupos: patrones GRASP<sup>3</sup>(patrones utilizados: alta cohesión, creador y controlador) y patrones GOF<sup>4</sup>(patrones utilizados: solitario, observador y solitario).

A continuación, se exponen algunos ejemplos de patrones que fueron tenidos en cuenta en el momento de desarrollar la propuesta.

- Patrón alta cohesión: El uso de este patrón se evidencia en la implementación de la clase Actions, la cual tiene la responsabilidad de definir las acciones para las plantillas y, paralelo a esto colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, expresado de otra manera, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas.

---

<sup>3</sup>GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades)

<sup>4</sup> GOF es un acrónimo de Gang of Four (La banda de los cuatro).

- Patrón creador: En las mencionadas clases se encuentran las acciones definidas para las operaciones lógicas del negocio en cuestión y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, esto evidencia que la clase Actions es "creadora" de dichas entidades.
- Patrón controlador: Todas las peticiones web, generalmente, son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación para un entorno determinado. El controlador es encargado de recibir la petición del usuario y en función de ella, envía la solicitud a las distintas clases para que sea procesada.
- Patrón creacional 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()`, la cual almacena una referencia a todos los objetos que forman el núcleo de Symfony y muy provechosamente, puede ser accedido desde cualquier punto de la aplicación.
- Patrón estructural decorator (envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo `layout.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora el contenido de la plantilla.
- Observer (Observador): se utiliza para definir una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Este patrón suele observarse en los frameworks de interfaces gráficas orientados a objetos, en los que la forma de capturar los eventos es suscribir 'listeners' a los objetos que pueden disparar eventos.

### **4.2.3. Vista lógica de la arquitectura de la solución**

La vista lógica es una abstracción de acuerdo a criterios diferentes que pueden usar su propia notación y definir de forma independiente el significado de los componentes, relaciones, argumentos, principios y pautas; es la base sobre la cual los arquitectos construyen modelos de aplicación que representan la perspectiva lógica de la arquitectura de una aplicación. Ver Anexo #2.

### 4.3. Diseño

El diseño es el flujo de trabajo que permite la comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, y tecnologías de interfaz de usuario y es el que da la entrada a las actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

#### 4.3.1. Modelo de Diseño

Según la definición Ivar Jacobson, Grady Booch y James Rumbaugh en su libro “El Proceso Unificado de Desarrollo de Software”: “Una clase de diseño y sus objetos, y de ese modo también los subsistemas que contienen las clases de diseño, a menudo participan en varias realizaciones de casos de uso. También puede darse el caso de algunas operaciones, atributos y asociaciones sobre una clase específica que son relevantes para sólo una realización de caso de uso. Esto es importante para coordinar todos los requisitos que diferentes realizaciones de casos de uso imponen a una clase, a sus objetos y a los subsistemas que contiene. Para manejar todo esto, utilizamos diagramas de clases conectados a una realización de caso de uso, mostrando sus clases participantes, subsistemas y sus relaciones. De esta forma podemos guardar la pista de los elementos participantes en una realización del caso de uso” (Jacobson, 2000).

Es en el modelo de diseño donde se definen las clases del diseño que conformarán el sistema que se va a implementar.

- Páginas clientes: Son las páginas encargadas de permitir a los usuarios interactuar con el sistema tanto para hacer solicitudes como para que sean mostradas las respuestas a las mismas.
- Páginas servidoras: Son las encargadas de la construcción de forma dinámica de las páginas clientes y sirven de enlace entre estas y el resto de las clases.
- Páginas controladoras: Son las responsables de realizar las operaciones que responden a los procesos de negocio y dar respuestas a las solicitudes hechas por el usuario.
- Clases entidad: Son las responsables de la persistencia de los datos físicamente.

Los diagramas de Clases del Diseño que componen el Modelo de Diseño se muestran en el Anexo # 3.

#### 4.4. Diseño de la Base de Datos

Los sistemas web que en la actualidad existen son desarrollos utilizando tecnologías cliente-servidor. Estos constan en la mayoría de los casos con bases de datos robustas que son las encargadas de almacenar todos los datos que en la misma se muestran. Producto de la complejidad y el cúmulo de información que se maneja se ha desarrollado una base de datos para almacenar toda la información que se controla dentro de cada subsistema, teniendo un diseño para cada uno. La misma viene incluida dentro de los anexos. (Ver Anexo # 4).

#### 4.5. Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. Consiste en:

- *Nodos*: Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- *Dispositivos*: Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- *Conectores*: Expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

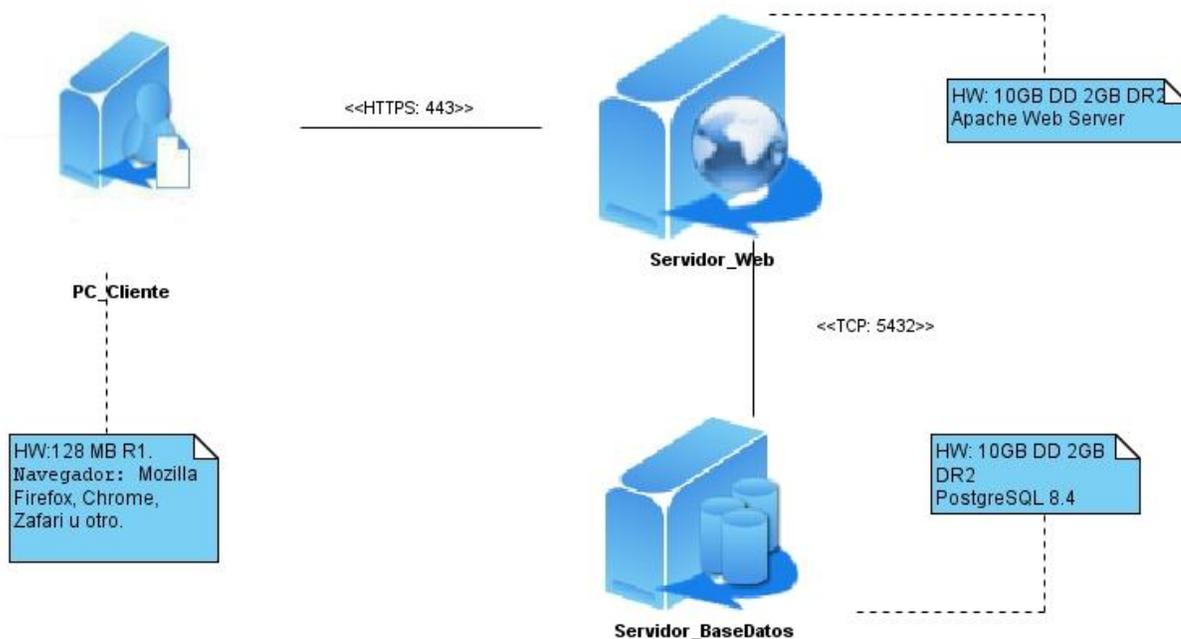


Ilustración 3 Modelo de Despliegue.

### Explicación del diagrama

En el modelo de despliegue que se presenta, se especifica la existencia por separado del servidor de aplicación y del servidor de bases de datos. Presenta además, el nodo correspondiente a la PC Cliente desde la cual accederán los especialistas para interactuar con la aplicación.

## 4.6. Modelo de Implementación.

### 4.6.1. Diagrama de Componentes

Un componente es una parte modular de un sistema, desplegable y reemplazable que típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.). Es un elemento de implementación que representa algo físico, ya sea ficheros o archivos, y son creados para poner el código, ya sea código fuente, código binario o código ejecutable. Y son establecidos, modificados o eliminados en el proceso de implementación. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Es un diagrama que muestra un conjunto de elementos del modelo tales

como componentes, subsistemas de implementación y sus relaciones. En el Anexo # 5 se muestra el diagrama de componentes del subsistema de Administración.

## 4.7. Pruebas

Siempre que se desarrolla algún producto es necesario probar lo que se hizo para comprobar que cumple con todo lo que se precisó en el inicio de su elaboración; el software no está ajeno a las pruebas y son estas un elemento crítico para la garantía de la calidad. Definitivamente la prueba de software representa una revisión final de las especificaciones del diseño y de la codificación. Los subsistemas que se desarrollaron también fueron sometidos a la fase de prueba, de los niveles de prueba se utilizó el nivel de prueba de sistema que son las pruebas que se realizan cuando el software está funcionando como un todo. Además, de los métodos de pruebas conocidos como caja blanca y caja negra se utilizó el método de caja negra que permite comprobar que cada funcionalidad es operativa. Dentro de las técnicas de prueba de caja negra se utilizó la técnica partición de equivalencia porque es una de las más efectivas ya que permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar.

### 4.7.1. Resultado

Fueron probados siete casos de uso, los que representan el cien por ciento del total de requisitos funcionales. Para cada caso de uso se tuvo en cuenta los diferentes escenarios que pudieran existir, los cuales implican cada una de las posibles interacciones del usuario o combinaciones de situaciones posibles con el fin de evaluar el comportamiento del sistema ante cada funcionalidad. El resultado de cada prueba coincide con la especificación de los requisitos lo que demuestra la correcta implementación de los mismos. Como ejemplo que muestra lo dicho anteriormente se refleja la sección y el escenario probado para el caso de uso Autenticar Usuario.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Autenticar usuario	EC 1.1: Autenticar usuario satisfactoriamente.	<ul style="list-style-type: none"> <li>• El actor escribe su nombre de usuario y contraseña y presiona el botón “Aceptar”.</li> <li>• El sistema valida los campos.</li> <li>• El sistema busca en la base de datos si ese usuario se encuentra registrado y su contraseña se corresponde con el usuario que intenta acceder.</li> <li>• El sistema verifica los permisos que tiene asignados y en dependencia de los mismos le permite acceder a la aplicación mostrando la interfaz correspondiente.</li> </ul>
	EC 1.2: Autenticar usuario falla.	<ul style="list-style-type: none"> <li>• Si el actor deja los campos en blanco y el sistema muestra un mensaje de error: “Los campos son obligatorios”.</li> <li>• Si el usuario no se corresponde con la contraseña enviada el sistema muestra un mensaje: “Usuario o contraseña incorrecto” El sistema da la posibilidad de volver a introducir los datos.</li> </ul>

Tabla 3: Diseño de Caso de prueba del CU: Autenticar usuario.

ID del escenario	Escenario	usuario	contraseña	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Autenticar usuario satisfactoriamente.	V	V	<ul style="list-style-type: none"> <li>• El sistema valida los campos.</li> <li>• El sistema busca en la base de datos si ese usuario pertenece y si su contraseña se corresponde con el usuario que intenta acceder.</li> <li>• El sistema verifica los permisos que tiene asignados y en dependencia de los permisos le permite acceder a la aplicación mostrando la interfaz.</li> </ul>	Resultado Satisfactorio.
EC 1.2	Autenticar usuario falla.	I	V	<ul style="list-style-type: none"> <li>• Si los datos de autenticación introducidos por el actor no son válidos, el sistema muestra el siguiente mensaje: "Usuario o contraseña incorrecta". El sistema da la posibilidad de volver a introducir los datos.</li> </ul>	Resultado Satisfactorio.

Tabla 4: Matriz de datos del CU: Autenticar usuario.

#### 4.8. Conclusiones Parciales

En el capítulo que recién culmina fueron tratados los aspectos más importantes en cuanto a la construcción del software. Dentro de estos aspectos se encuentra la arquitectura que forma parte de una de las etapas fundamentales del desarrollo de software que junto con la definición de los patrones de diseño que serían utilizados, permitió a todo el equipo de desarrollo trabajar en una línea común para lograr alcanzar los objetivos trazados, cubriendo así todas las necesidades planteadas. También se estableció el modelo entidad - relación que utilizarían los dos módulos lo que permitió establecer las clases persistentes, sus atributos, relaciones y propiedades.

También fue necesario realizar el modelo de despliegue lo cual garantiza que se pueda tener una idea de cómo estará distribuido físicamente el sistema para su correcto funcionamiento. Luego de tener los dos módulos implementados fue necesario someterlos a la fase de prueba la cuál se encargaría de comprobar que los mismos cumplen con todos los requisitos establecidos en el inicio de su elaboración.

## CONCLUSIONES GENERALES.

Una vez concluida la presente investigación en la cual se realizó la implementación de los módulos de administración y gestión de la información para el SIG INRH, se puede arribar a las siguientes conclusiones:

- La caracterización de los procesos de administración y gestión de la información de los SIG aplicados a los objetivos hidráulicos cubanos permitió que se tuvieran los elementos fundamentales a tener en cuenta para la construcción de los dos módulos. También la descripción de los conceptos asociados al tema posibilitaron una mejor comprensión del objeto de estudio. Otro rasgo significativo fue el estudio que se hizo a nivel nacional e internacional de soluciones similares; el cuál arrojó como resultado que ninguna de las referenciadas cumplían con los requisitos específicos que persiguen los dos módulos para el SIG INRH.
- La herramienta, metodología y lenguaje de modelado identificado a partir de la investigación de las tendencias y tecnologías actuales para el desarrollo de software garantizó la realización del correcto modelado de un conjunto de artefactos necesarios para la implementación de la aplicación. Por su parte, la investigación de los lenguajes de programación, herramientas y tecnologías que serían utilizadas para la construcción de la solución arrojó que serían utilizados los de última generación que ayudarían a la construcción de un software que estuviera al nivel de las últimas tendencias.
- La definición de la arquitectura que sería utilizada permitió obtener una visión de cómo estaría estructurado el sistema y de cómo sus componentes trabajarían juntos. El uso del estilo de llamada y retorno permitió conseguir una estructura relativamente fácil de modificar y escalar. Unido a esto la utilización de los patrones de diseño para la construcción de los diagramas de las clases del diseño garantizaron diseñar un software bien orientado a objetos. Estos últimos permitieron además que existiera poca dependencia entre las clases permitiendo así una mejor reutilización.
- La validación de la solución construida, a través del uso del método de caja negra arrojó que los dos módulos cumplían con las condiciones y cualidades especificadas en su conceptualización, trayendo como beneficio la satisfacción total del cliente y del equipo de desarrollo.

Por todo lo antes mencionado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

## **RECOMENDACIONES**

- Se recomienda implementarle al módulo de administración la funcionalidad de llevar el registro de todas las acciones realizadas por un usuario dentro del sistema, para una mejor administración de la información que se manipula.
- Se recomienda que el módulo de administración sea utilizado por el resto de los SIG que se desarrollan por el proyecto Aplicativos SIG.
- Se recomienda que el módulo de gestión de la información dé la posibilidad de modificar los datos de un recurso hidráulico cuando este recurso sea seleccionado en el mapa.

## REFERENCIAS BIBLIOGRÁFICAS

1. 27000, ISO. Patente 27000. Mayo de 2009.
2. (IRIS-CERT, 2008) Autores. «Conferencia 2. Seguridad Informática.» Control de acceso. Identificación y autenticación. La Habana, 2008.
3. «Conferencia 3. Seguridad Informática.» Ataques informáticos y mecanismos de defensa. La Habana, 2006.
4. Chang, Kai Chen. «Introduction to Geographic Information Systems.» McGraw-Hill, 2006.
5. DDC, Autores. «Conferencia 1. Seguridad Informática.» Introducción a la Seguridad Informática. La Habana, 2010.
6. Definición.de. Definición.de. 2008. <http://definicion.de/administracion/> (último acceso: 29 de 11 de 2010).
7. Lewis, Christian. Administración de Aplicaciones en ambientes Web. 19 de 08 de 2009. <http://www.tecnologiahechapalabra.com/datos/software/articulo.asp?i=4036>.
8. Marañón, Gonzalo Álvarez. Autenticación y Autorización. 2002. <http://www.iec.csic.es/cryptonomicon/autenticacion/cert.html> (último acceso: 30 de Noviembre de 2010).
9. Öztürk, Ahmet. Computing Information Services Newsletter. julio de 2002. <http://cism.metu.edu.tr/2002-6/free.php> (último acceso: 29 de noviembre de 2010).
10. Rodríguez-Tastets, M. Andrea. «Spatial Databases.» Spatial Databases. Chile, 2008.
11. Hidráulicos, I. N. (s.f.). Recursos Hidráulicos. Recuperado el 13 de Noviembre de 2010, de [http://www.cubagob.cu/des\\_eco/inrh/rec\\_hid.htm](http://www.cubagob.cu/des_eco/inrh/rec_hid.htm).
12. TICS [En línea]. - 21 de noviembre de 2010. - [http://tics.org.ar/index.php?option=com\\_content&task=view&id=13&Itemid=28](http://tics.org.ar/index.php?option=com_content&task=view&id=13&Itemid=28).
13. Mapping Interactivo. Revista Internacional de Ciencias de la Tierra. [En línea] / aut. Titular Dr. José Luis Batista Silva Investigador. - Noviembre de 2005. - 21 de Noviembre de 2010. - [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1051](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051).
14. Camacho, E., Cardeso, F., & Núñez, g. (abril de 2004). Arquitecturas de Software.
15. IRIS-CERT. (2008). RedIRIS. Recuperado el 18 de noviembre de 2011, de <http://www.rediris.es/cert/doc/unixsec/node14.html>
16. Sotelo 2008, M. R. (s.f.). Mygnet. Recuperado el 18 de noviembre de 2011, de <http://www.mygnet.com/articulos/seguridad/763/index.php>

17. Herrera Valencia, Joan Ignasi, Pereira Cao, Mercè e Insua de la Fuente, Carlos. 2008. Infraestructuras de datos para la localización y acceso inteligente a la información disponible a través de servicios web de catálogo. 2008.
18. DESARROLLOWEB. "Características de PHP5". Disponible en: <http://www.desarrolloweb.com/articulos/1699.php>.
19. Barcelona), Pau Serra del Pozo. Oficina Técnica de Cartografía y SIG Local (Diputación de «Mappingin.» Octubre de 2002. [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=179](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=179) (último acceso: 06 de febrero de 2011).
20. Potencier, F., & Zaninotto, F. (2008). Symfony guía definitiva. [www.librosweb.es](http://www.librosweb.es).
21. Hinostraza, R. R. (2007) "Características de PHP." Volumen, DOI:
22. Oficial. (Bibliografía). Visual-paradigm. Recuperado el 25 de febrero de 2011, de <http://www.visual-paradigm.com/>
23. Ivar Jacobson, G. B. y J. R. El Proceso Unificado de Desarrollo de Software. Madrid, 2000a. P.
24. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo del Software. Madrid : Addison Wesley, 2000. pág. Capítulo 6. 28006.
25. Propuesta de diseño para proyectos informáticos que utilizan Symfony como Framework. Martell, Vladimir, Vega, Yurisbel y Figueroa, Daira. 2010. 5, La Habana : publicaciones. uci.cu, 2010, Vol. 3.
26. Ravi, S y U.J.D. 1986. Compilers: principles, techniques, and tools. Boston. : Addison-Wesley Longman Publishing Co., 1986. pág. Citado en p. 14.

## BIBLIOGRAFÍA

1. Chang, Kai Chen. «Introduction to Geographic Information Systems.» *McGraw-Hill*, 2006.
2. Date, C. J. *Introducción a los Sistemas de Bases de Datos*. México: Preason Educacion, 2001.
3. Definición.de. *Definición.de*. 2008. <http://definicion.de/administracion/> (último acceso: 29 de 11 de 2010).
4. *GNU Operatin System*. (s.f.). Recuperado el 13 de Noviembre de 2010, de <http://www.gnu.org/philosophy/free-sw.es.html>
5. Hidráulicos, I. N. (s.f.). *Recursos Hidráulicos*. Recuperado el 13 de Noviembre de 2010, de [http://www.cubagob.cu/des\\_eco/inrh/rec\\_hid.htm](http://www.cubagob.cu/des_eco/inrh/rec_hid.htm).
6. Hinostroza, R. R. (2007) "Características de PHP." *Volumen*, DOI:
7. INRH. (s.f.). *CubAgua*. Obtenido de <http://www.hidro.cu/>
8. Ivar Jacobson, G. B. y J. R. El Proceso Unificado de Desarrollo de Software. Madrid, 2000a. P.
9. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo del Software. Madrid : Addison Wesley, 2000. pág. Capítulo 6. 28006.
10. Mapping Interactivo. Revista Internacional de Ciencias de la Tierra. [En línea] / aut. Titular Dr. José Luis Batista Silva Investigador. - Noviembre de 2005. - 21 de Noviembre de 2010. - [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1051](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051).
11. *Microsoft-Technet*. (s.f.). Recuperado el 13 de Noviembre de 2010, de <http://technet.microsoft.com/es-es/library/bb964711.aspx>
12. Modificado de Bosh, Jan. Design and Use of Software Architectures, Adopting and Evolving a Product-Line Approach, first edition, London: Addison Wesley, 2000.
13. «NetBeans.» <http://netbeans.org/community/releases/65/> (último acceso: 16 de febrero de 2011).
14. Oficial. (Bibliografía). Visual-Paradigm. Recuperado el 25 de febrero de 2011, de <http://www.visual-paradigm.com/>
15. Otros, R. y. (1998). *Sistemas de Información Geográfica*.
16. Potencier, F., & Zaninotto, F. (2008). Symphony guía definitiva. [www.librosweb.es](http://www.librosweb.es).

17. Propuesta de diseño para proyectos informáticos que utilizan Symfony como Framework. Martell, Vladimir, Vega, Yurisbel y Figueroa, Daira. 2010. 5, La Habana : publicaciones. uci.cu, 2010, Vol. 3. 5.
18. Ravi, S y U.J.D. 1986. Compilers: principles, techniques, and tools. Boston. : Addison-Wesley Longman Publishing Co., 1986. pág. Citado en p. 14.
19. Rodríguez, C. (2008). *Transformaciones tecnológicas* . La Habana.
20. SENCHA. «SENCHA.» <http://www.sencha.com/> (último acceso: 06 de 02 de 2011).
21. SIG-ESAC: Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba [Publicación periódica] / aut. Núñez MSc. Héctor Manuel Fernández // Revista Cubana de Higiene y Epidemiología. - 2006.
22. *SoftwareLibre*. (s.f.). Recuperado el 13 de Noviembre de 2010, de <http://www.softwarelibre.org/>
23. TICS [En línea]. - 21 de noviembre de 2010. - [http://tics.org.ar/index.php?option=com\\_content&task=view&id=13&Itemid=28](http://tics.org.ar/index.php?option=com_content&task=view&id=13&Itemid=28).
24. Traducido de Clements, P.; Bass, L.; Kazman, R. Software Architecture in Practice. Edited by SEI Series, first edition, SEI Series in Software Engineering: Addison Wesley, 2003.
25. Traducido y modificado de White, Sharon A.; Lemus-Olalde, Cuauhtémoc. The Software Architecture Process, 1997.
26. *Wikipedia-Software Libre*. (2001). Recuperado el 13 de Noviembre de 2010, de [http://es.wikipedia.org/wiki/Software\\_libre](http://es.wikipedia.org/wiki/Software_libre)
27. Wikipedia. (Octubre de 2005). *Datos Espaciales*. Recuperado el 13 de Noviembre de 2010, de Wikipedia: [http://es.wikipedia.org/wiki/Base\\_de\\_datos\\_especial](http://es.wikipedia.org/wiki/Base_de_datos_especial)
28. Wikipedia. (23 de Octubre de 2003). *Sistema de Información Geográfica*. Recuperado el 13 de Noviembre de 2010, de Wikipedia: [http://es.wikipedia.org/wiki/Sistema\\_de\\_Informaci%C3%B3n\\_Geogr%C3%A1fica](http://es.wikipedia.org/wiki/Sistema_de_Informaci%C3%B3n_Geogr%C3%A1fica)

## ANEXOS

### Anexo 1 Especificación textual de Caso de Uso.

<b>Caso de Uso:</b>	Gestionar Roles
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar, eliminar o insertar un rol al sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario administrador desea insertar, modificar, asignar permisos o eliminar un rol y termina cuando haya sido ejecutada alguna de las acciones anteriormente especificadas.
<b>Precondiciones:</b>	El usuario tiene que estar autenticado.
<b>Referencias</b>	RF 8, RF 9, RF 10, RF 11, RF 12.
<b>Prioridad</b>	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El caso de uso se inicia cuando el usuario desea realizar una de las opciones que se especifican a continuación: (Ver Interfaz 1)</p> <ul style="list-style-type: none"> <li>- Listar Roles.</li> <li>- Adicionar Rol.</li> <li>- Modificar Rol.</li> <li>- Asignar Permiso.</li> <li>- Eliminar Rol.</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Listar Roles”, ver sección “Listar Roles”,.</li> <li>- Si selecciona “Adicionar Rol”, ver sección “Adicionar Rol”.</li> <li>- Si selecciona “Modificar Rol”, ver sección “Modificar Rol”.</li> <li>- Si selecciona “Asignar Permisos”, ver sección “Asignar Permisos”.</li> <li>- Si selecciona “Eliminar Rol”, ver sección</li> </ul>

	“Eliminar Rol”.
	3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario.

**Prototipo de Interfaz**

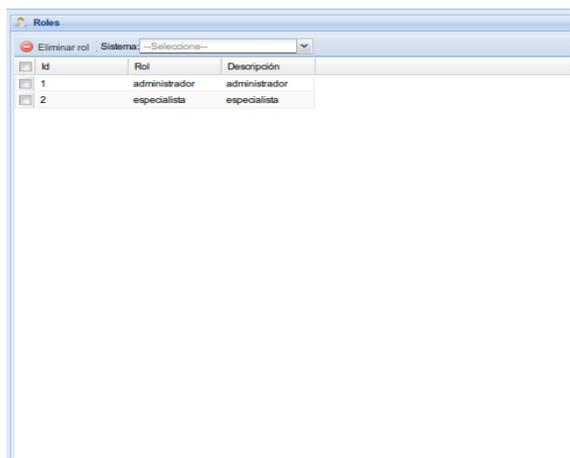
**Interfaz 1**



**Sección “Listar Roles”**

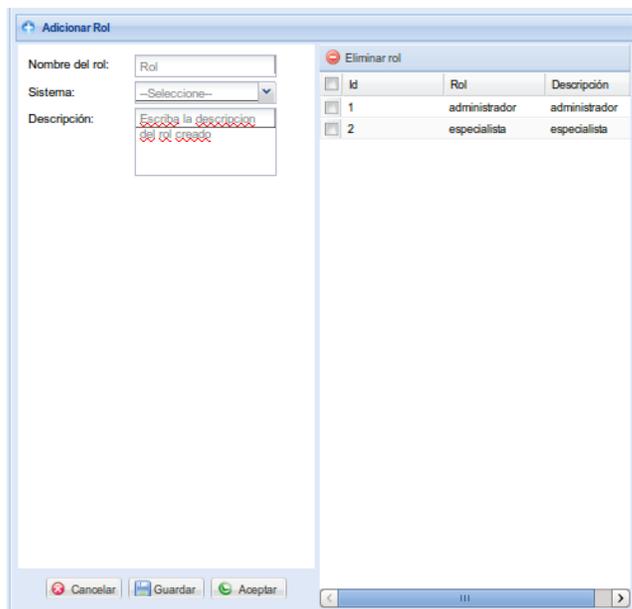
Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Listar Usuario (Ver Interfaz 1)	4. El sistema comprueba que existan roles registrados y procede a mostrar una tabla que contiene el listado de todos los roles que se encuentran añadidos al sistema. (Ver Interfaz 2)

**Interfaz 2**



<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>4.1</b> El sistema muestra un mensaje de error indicando que no hay roles añadidos al sistema finalizando así el caso de uso.
<b>Sección “Adicionar Rol”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>3.</b> El usuario selecciona la opción Adicionar Rol. (Ver Interfaz 1)	<b>4.</b> El sistema muestra un formulario que le permite al usuario insertar los datos (Rol y Descripción). (Ver Interfaz 3)
<b>5.</b> El usuario inserta los datos (Rol y Descripción).y da clic en el botón guardar.	<b>6.</b> El sistema valida que todos los campos estén correctos.
	<b>7.</b> El sistema verifica que el rol no se encuentre insertado.
	<b>8.</b> El sistema guarda el nuevo rol manteniendo la interfaz de Adicionar Rol finalizando así el caso de uso.

Interfaz 3

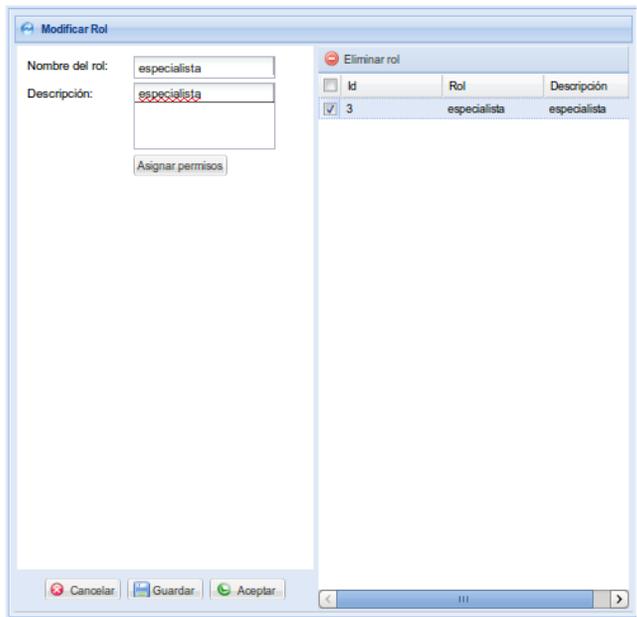


Flujo Alterno

Acción del Actor	Respuesta del Sistema
<p><b>5.1</b> El usuario inserta los datos (Rol y Descripción) y da clic en el botón Aceptar.</p>	<p><b>8.1</b> Realiza los pasos 6 y 7 del flujo normal de los eventos, guarda el rol y procede a cerrar la interfaz de Adicionar Rol, finalizando así el caso de uso.</p>
	<p><b>6.1</b> El sistema le informa al usuario que hay error en la entrada de los datos y el usuario vuelve al paso 3 del flujo normal de los eventos.</p>
	<p><b>7.1</b> El sistema envía un mensaje de error especificando que el usuario se encuentra registrado en el sistema, y vuelve al paso 3 si desea adicionar otro rol, en caso contrario finaliza el caso de uso.</p>

Sección “Modificar Rol.”	
Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Modificar Rol (Ver Interfaz 1)	4. Ir al paso 2 del flujo normal de los eventos de la sección Listar Roles.
5. El usuario selecciona el rol que desea modificar. (Ver Interfaz 4)	6. El sistema llena el formulario con los datos que tiene guardado del usuario seleccionado.
7. El usuario modifica los datos (Rol y Descripción) y da clic en el botón Guardar.	8. El sistema valida que todos los campos estén correctos.
	9. El sistema guarda los cambios realizados por el usuario, manteniendo la interfaz de Modificar Rol, finalizando así el caso de uso.

Interfaz 4



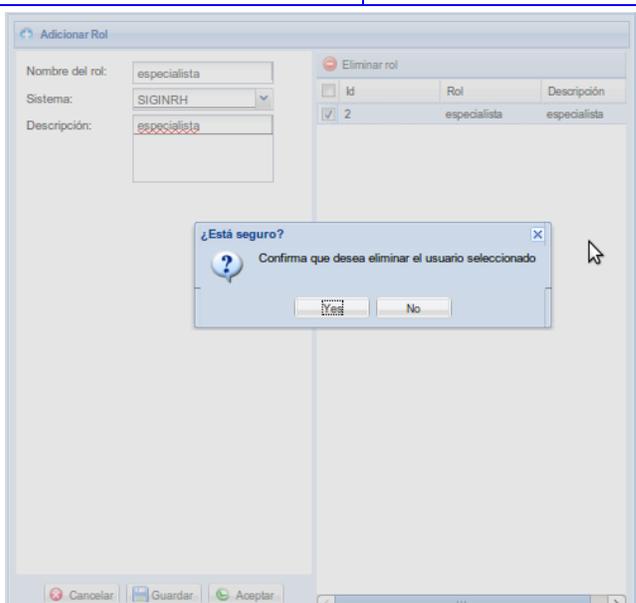
Flujo Alterno

Acción del Actor	Respuesta del Sistema
7.1. El usuario modifica los datos (Rol y Descripción) y da clic en el botón Aceptar.	9.1. El sistema realiza el paso 8 del flujo normal de eventos y procede a cerrar la interfaz de Modificar Rol, finalizando así el caso de uso.
	8.1. El sistema muestra un mensaje indicando que hay error en la entrada de los datos y vuelve al paso 4.

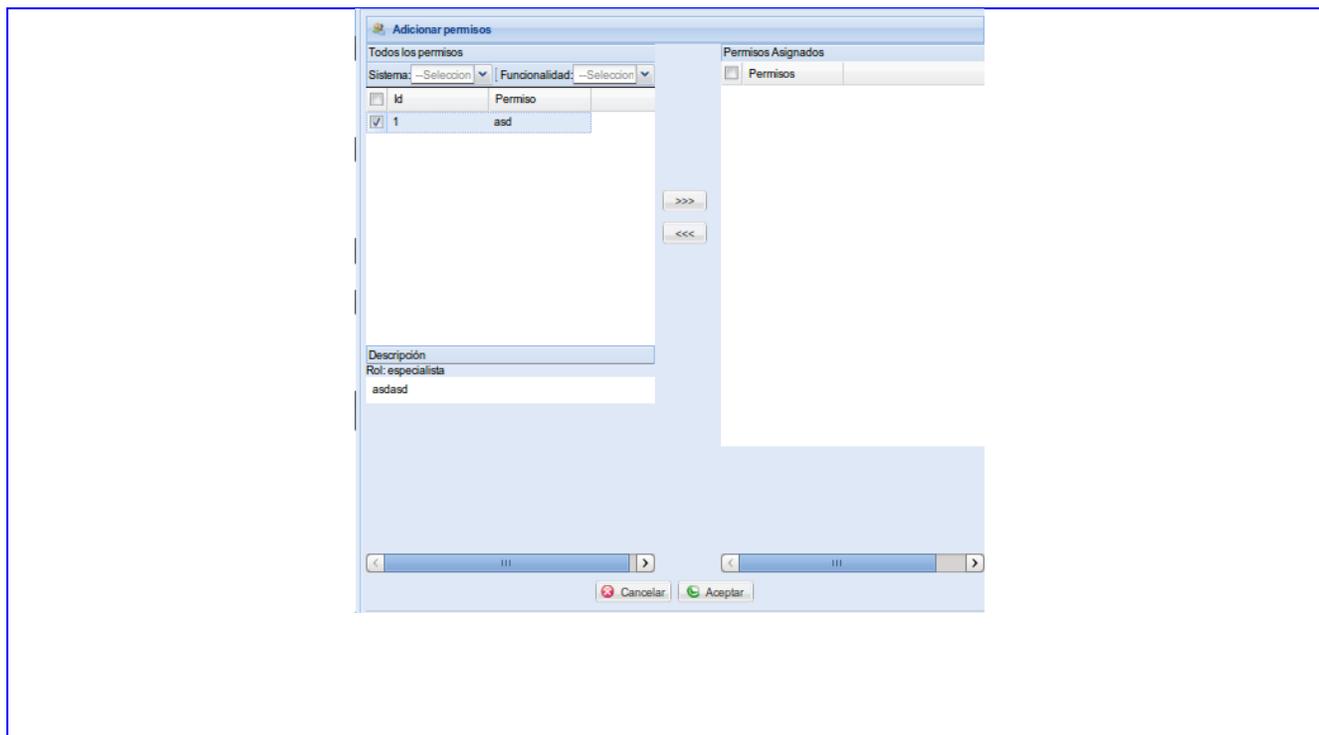
**Sección “Eliminar Rol”**

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona alguna de las opciones disponibles desde la interfaz 1. (Modificar Rol o Listar Rol)	4. El sistema ejecuta el paso 4 del flujo normal de los eventos de las secciones Modificar Rol o Listar Rol.
5. El usuario selecciona al usuario que desea eliminar y da clic en la opción Eliminar Rol.	6. El sistema muestra un mensaje al usuario pidiéndole confirmación de la acción que desea realizar. (Ver Interfaz 5)
7. El usuario confirma la acción.	8. El sistema elimina el rol seleccionado por el usuario y finaliza el caso de uso.

**Interfaz 5**



<b>Flujo Alterno.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5.1. El usuario no selecciona ningún rol y da clic en la opción Eliminar Rol	6.1 El sistema muestra un mensaje donde se especifica que debe seleccionar un rol.
7.1 El usuario no confirma la acción.	8.1. El sistema no elimina el rol seleccionado por el usuario y finaliza el caso de uso.
<b>Sección “Asignar Permisos”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El usuario selecciona la opción Modificar Rol.	4. El sistema muestra la interfaz Modificar Rol.
5. El usuario selecciona el rol que le desea reajustar los permisos y da clic en la opción Cambiar Permisos.	6. El sistema muestra la interfaz Asignar Permisos. (Ver Interfaz 6)
7. El usuario selecciona un sistema de la lista de permisos que no le han sido asignados al rol seleccionado y da doble clic.	8. El sistema guarda el permiso seleccionado en la lista de permisos añadidos al rol.
9. El usuario da clic en el botón Aceptar.	10. El sistema cierra esta interfaz y vuelve a la interfaz de Modificar Rol.
<b>Interfaz 6</b>	



Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	<b>6.1.</b> El sistema muestra un mensaje indicando que debe seleccionar un rol para poder asignarle sistemas.
Postcondiciones	El sistema inserta, elimina, modifica, le asigna permisos, a los roles a partir de la opción que se desee realizar.

Tabla 5: Descripción textual del CU Gestionar Roles

Tabla 6: Descripción de CU Gestionar Sistema.

<b>Caso de Uso:</b>	Gestionar Sistema.
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar, eliminar o insertar un sistema del sistema.

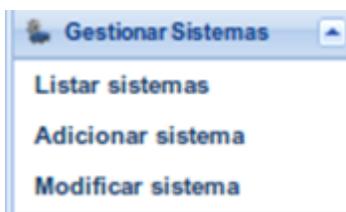
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario administrador desea insertar, modificar o eliminar un sistema y termina cuando haya sido ejecutada alguna de las acciones anteriormente especificadas.
<b>Precondiciones:</b>	El usuario tiene que estar autenticado.
<b>Referencias</b>	RF 13, RF 14, RF 15, RF 16.
<b>Prioridad</b>	Crítico

**Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p><b>1.</b> El caso de uso se inicia cuando el usuario desea realizar una de las opciones que se especifican a continuación: (Ver Interfaz 1)</p> <ul style="list-style-type: none"> <li>- Listar Sistemas.</li> <li>- Adicionar Sistema.</li> <li>- Modificar Sistema.</li> <li>- Eliminar Sistema.</li> </ul>	<p><b>2.</b> El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Listar Sistemas”, ver sección “Listar Sistemas”, ”.</li> <li>- Si selecciona “Adicionar Sistema”, ver sección “Adicionar Sistema”.</li> <li>- Si selecciona “Modificar Sistema”, ver sección “Modificar Sistema”.</li> <li>- Si selecciona “Eliminar Sistema”, ver sección “Eliminar Sistema”.</li> </ul>
	<p><b>3.</b> El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario.</p>

**Prototipo de Interfaz**

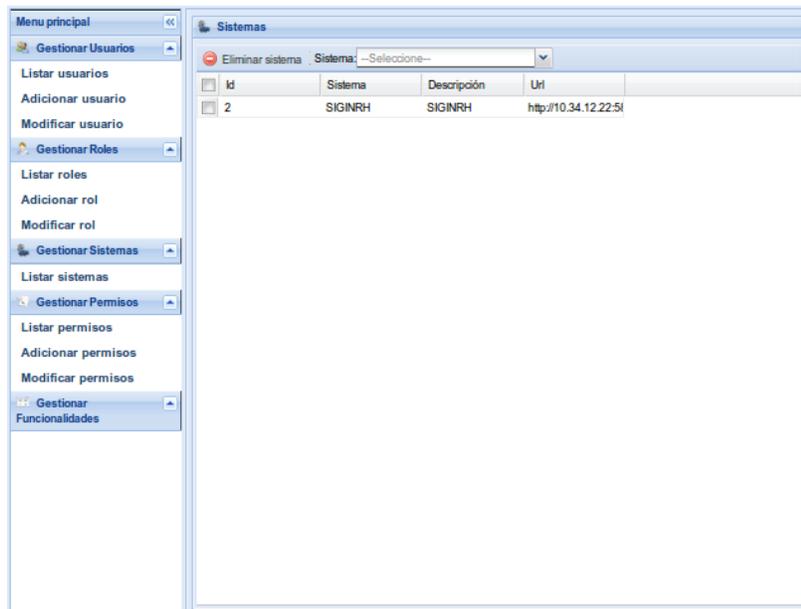
**Interfaz 1**



**Sección “Listar Sistemas”**

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Listar Sistema (Ver Interfaz 1)	4. El sistema comprueba que existan sistemas registrados y procede a mostrar una tabla que contiene el listado de todos los sistemas que se encuentran añadidos al sistema. (Ver Interfaz 2)

**Interfaz 2**

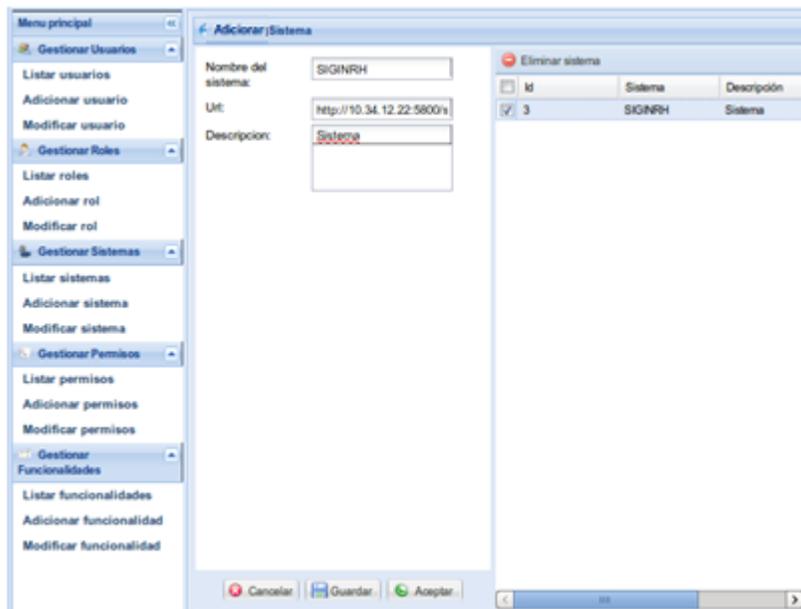


**Flujos Alternos**

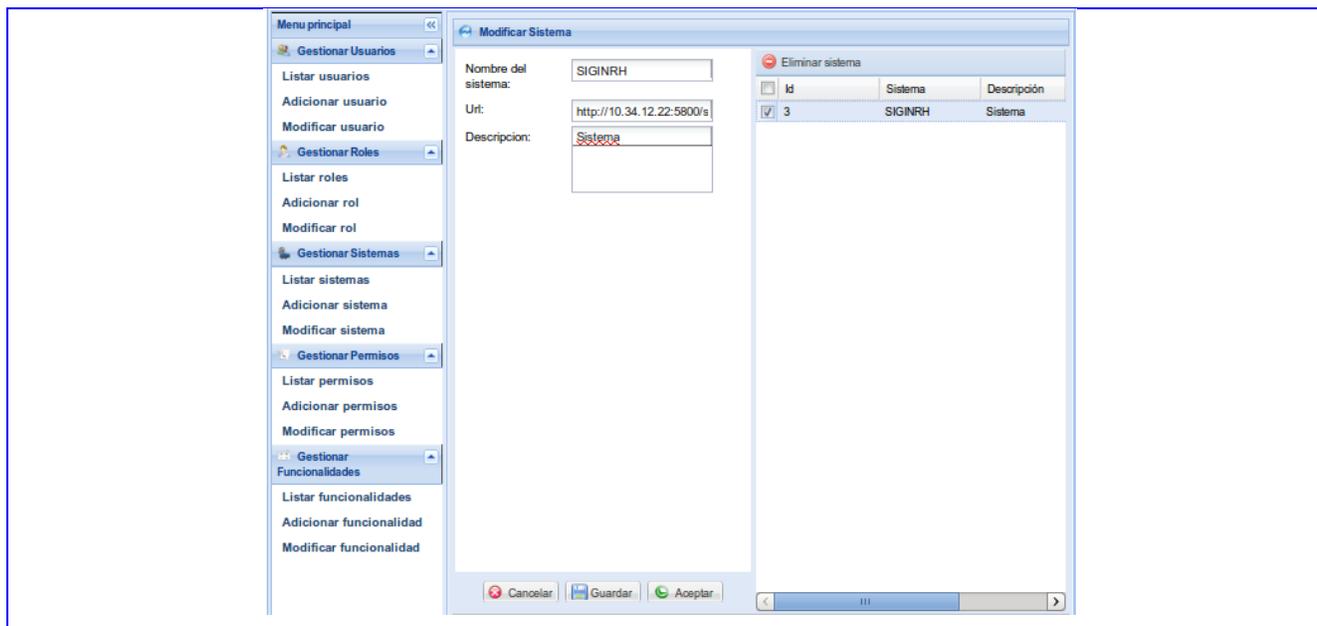
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje de error indicando que no hay sistemas añadidos al sistema finalizando así el caso de uso.

Sección "Adicionar Sistema"	
Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Adicionar Sistema. (Ver Interfaz 1)	4. El sistema muestra un formulario que le permite al usuario insertar los datos (Sistema, URL y Descripción).
5. El usuario inserta los datos (Sistema, URL y Descripción).y da clic en el botón Guardar.	6. El sistema valida que todos los campos estén correctos.
	7. El sistema verifica que el sistema no se encuentre insertado.
	8. El sistema guarda el nuevo sistema manteniendo la interfaz de Adicionar Sistema finalizando así el caso de uso.

Interfaz 3



<b>Flujo Alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
5.1 El usuario inserta los datos (Sistema, URL y Descripción) y da clic en el botón Aceptar.	8.1 Realiza los pasos 6 y 7 del flujo normal de los eventos, guarda el sistema y procede a cerrar la interfaz de Adicionar Sistema, finalizando así el caso de uso.
	6.1 El sistema le informa al usuario que hay error en la entrada de los datos y el usuario vuelve al paso 3 del flujo normal de los eventos.
	7.1 El sistema envía un mensaje de error especificando que el usuario se encuentra registrado en el sistema, y vuelve al paso 3 si desea adicionar otro sistema, en caso contrario finaliza el caso de uso.
<b>Sección “Modificar Sistema.”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El usuario selecciona la opción Modificar Sistema (Ver Interfaz 1)	4. Ir al paso 2 del flujo normal de los eventos de la sección Listar Sistemas.
5. El usuario selecciona el sistema que desea modificar. (Ver Interfaz 4)	6. El sistema llena el formulario con los datos que tiene guardado del usuario seleccionado.
7. El usuario modifica los datos (Sistema, URL y Descripción) y da clic en el botón guardar.	8. El sistema valida que todos los campos estén correctos.
	9. El sistema guarda los cambios realizados por el usuario, manteniendo la interfaz de Modificar Sistema, finalizando así el caso de uso.
<b>Interfaz 4</b>	



**Flujo Alterno**

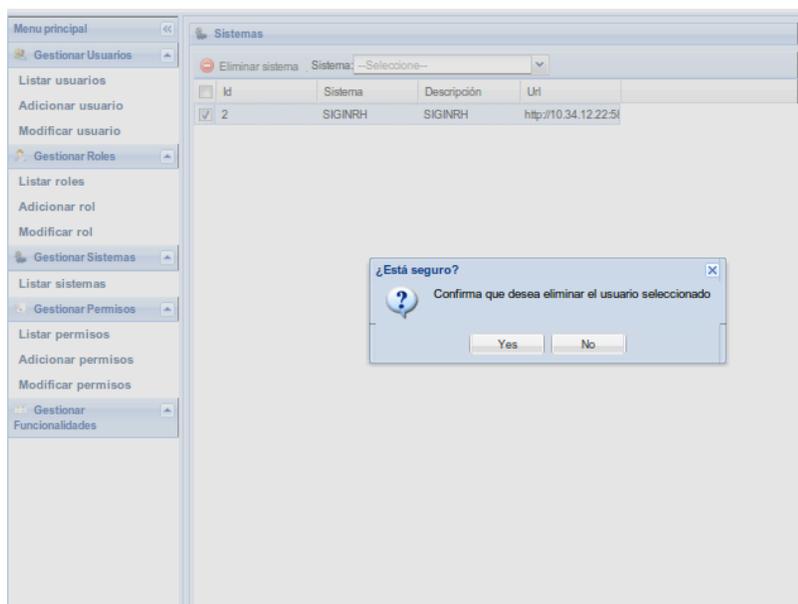
Acción del Actor	Respuesta del Sistema
7.1. El usuario modifica los datos (Sistema, URL y Descripción) y da clic en el botón aceptar.	9.1. El sistema realiza el paso 8 del flujo normal de eventos y procede a cerrar la interfaz de Modificar Sistema, finalizando así el caso de uso.
	8.1. El sistema muestra un mensaje indicando que hay error en la entrada de los datos y vuelve al paso 4.

**Sección “Eliminar Sistema”**

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona alguna de las opciones disponibles desde la interfaz 1. (Modificar Sistema o Listar Sistema)	4. El sistema ejecuta el paso 4 del flujo normal de los eventos de las secciones Modificar Sistema o Listar Sistema.
5. El usuario selecciona al usuario que desea eliminar y da clic en la opción Eliminar Sistema.	6. El sistema muestra un mensaje al usuario pidiéndole confirmación de la acción que desea

	realizar.
7. El usuario confirma la acción.	8. El sistema elimina al sistema seleccionado por el usuario y finaliza el caso de uso.

**Interfaz 5**



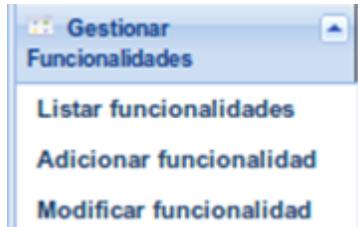
**Flujo Alternativo.**

Acción del Actor	Respuesta del Sistema
5.1. El usuario no selecciona ningún sistema y da clic en la opción Eliminar Sistema	6.1 El sistema muestra un mensaje donde se especifica que debe seleccionar un sistema.
7.1 El usuario no confirma la acción.	8.1. El sistema no elimina al sistema seleccionado por el usuario y finaliza el caso de uso.
<b>Postcondiciones</b>	El sistema inserta, elimina, modifica, le asigna permisos, a los sistemas a partir de la opción que se desee realizar.

Tabla 7: Descripción textual del CU Gestionar Acciones.

<b>Caso de Uso:</b>	Gestionar Acciones
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar, eliminar o insertar una funcionalidad al sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario administrador desea insertar, modificar o eliminar una funcionalidad y termina cuando haya sido ejecutada alguna de las acciones anteriormente especificadas.
<b>Precondiciones:</b>	El usuario tiene que estar autenticado.
<b>Referencias</b>	RF 17, RF 18, RF 19, RF 20.
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el usuario desea realizar una de las opciones que se especifican a continuación: (Ver Interfaz 1)</p> <ul style="list-style-type: none"> <li>- Listar Acciones.</li> <li>- Adicionar Funcionalidad.</li> <li>- Modificar Funcionalidad.</li> <li>- Eliminar Funcionalidad.</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Listar Acciones”, ver sección “Listar Acciones”, ”.</li> <li>- Si selecciona “Adicionar Funcionalidad”, ver sección “Adicionar Funcionalidad”.</li> <li>- Si selecciona “Modificar Funcionalidad”, ver sección “Modificar Funcionalidad”.</li> <li>- Si selecciona “Eliminar Funcionalidad”, ver sección “Eliminar Funcionalidad”.</li> </ul>
	<p>3. El caso de uso termina cuando el sistema procesa la información según la funcionalidad realizada por el usuario.</p>
<b>Prototipo de Interfaz</b>	

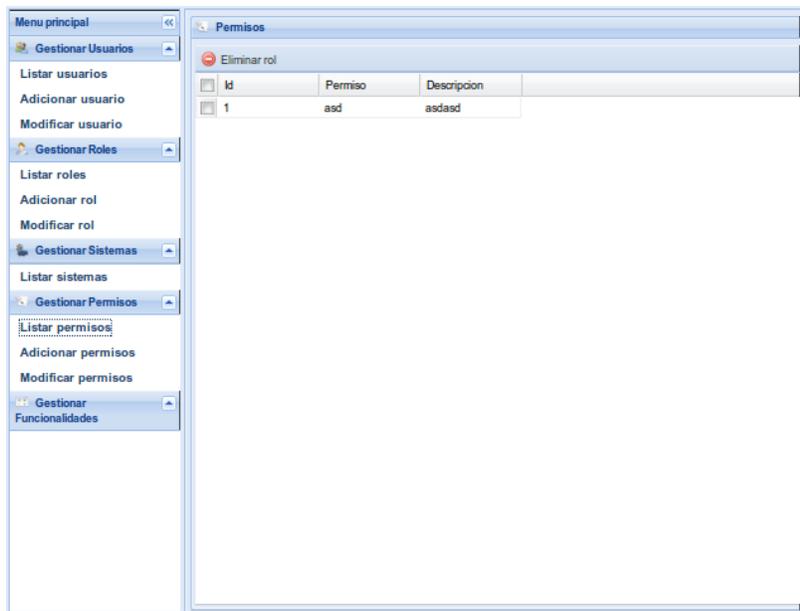
**Interfaz 1**



**Sección “Listar Acciones”**

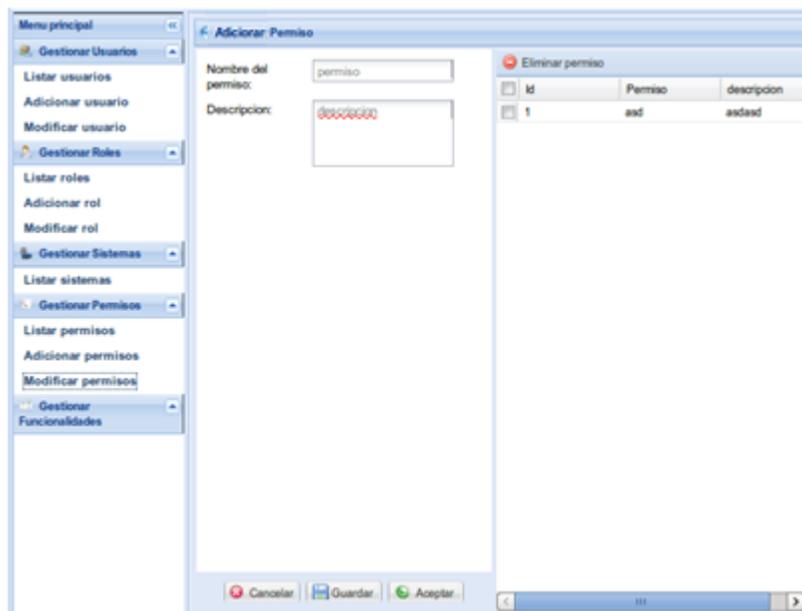
Acción del Actor	Respuesta del Sistema
<p>3. El usuario selecciona la opción Listar Funcionalidad (Ver Interfaz 1)</p>	<p>4. El sistema comprueba que existan acciones registrados y procede a mostrar una tabla que contiene el listado de todas las acciones que se encuentran añadidos al sistema. (Ver Interfaz 2)</p>

**Interfaz 2**



<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>4.1</b> El sistema muestra un mensaje de error indicando que no hay acciones añadidas al sistema finalizando así el caso de uso.
<b>Sección “Adicionar Acción”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>3.</b> El usuario selecciona la opción Adicionar Acción. (Ver Interfaz 1)	<b>4.</b> El sistema muestra un formulario que le permite al usuario insertar los datos (Acción y Descripción).
<b>5.</b> El usuario inserta los datos (Funcionalidad y Descripción).y da clic en el botón guardar.	<b>6.</b> El sistema valida que todos los campos estén correctos.
	<b>7.</b> El sistema verifica que la Funcionalidad no se encuentre insertada.
	<b>8.</b> El sistema guarda la nueva Funcionalidad manteniendo la interfaz de Adicionar Funcionalidad finalizando así el caso de uso.

Interfaz 3



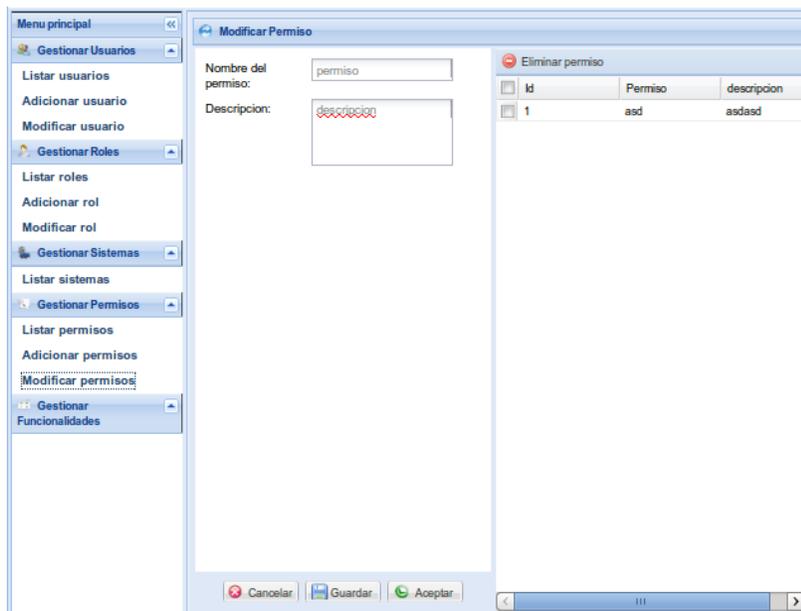
Flujo Alterno

Acción del Actor	Respuesta del Sistema
5.1 El usuario inserta los datos (Funcionalidad y Descripción) y da clic en el botón Aceptar.	8.1 Realiza los pasos 6 y 7 del flujo normal de los eventos, guarda la Funcionalidad y procede a cerrar la interfaz de Adicionar Funcionalidad, finalizando así el caso de uso.
	6.1 El sistema le informa al usuario que hay error en la entrada de los datos y el usuario vuelve al paso 3 del flujo normal de los eventos.
	7.1 El sistema envía un mensaje de error especificando que el usuario se encuentra registrado en el sistema, y vuelve al paso 3 si desea adicionar otra acción, en caso contrario finaliza el caso de uso.

Sección “Modificar Acción.”

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Modificar Funcionalidad (Ver Interfaz 1)	4. Ir al paso 2 del flujo normal de los eventos de la sección Listar Acciones.
5. El usuario selecciona la Funcionalidad que desea modificar. (Ver Interfaz 4)	6. El sistema llena el formulario con los datos que tiene guardado del usuario seleccionado.
7. El usuario modifica los datos (Funcionalidad y Descripción) y da clic en el botón Guardar.	8. El sistema valida que todos los campos estén correctos.
	9. El sistema guarda los cambios realizados por el usuario, manteniendo la interfaz de Modificar Funcionalidad, finalizando así el caso de uso.

Interfaz 4



Flujo Alterno

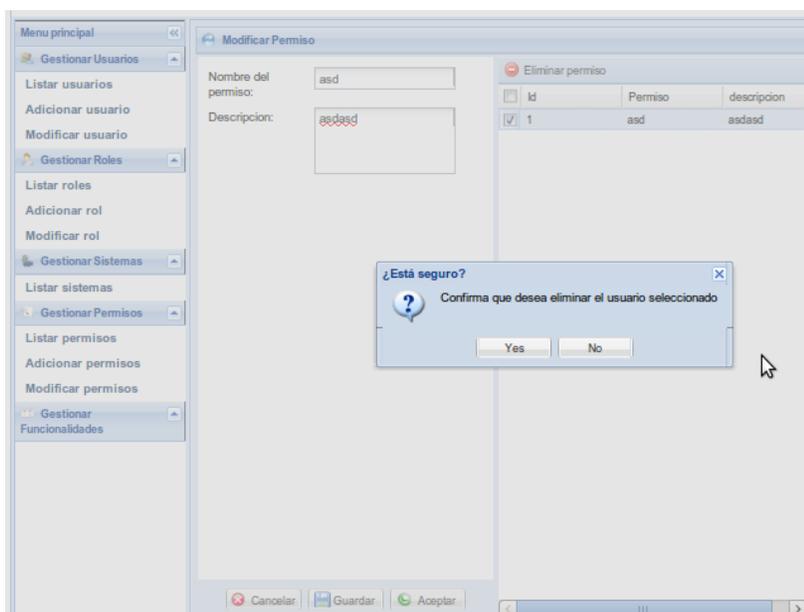
Acción del Actor	Respuesta del Sistema
7.1. El usuario modifica los datos (Funcionalidad	9.1. El sistema realiza el paso 8 del flujo normal

y Descripción) y da clic en el botón aceptar.	de eventos y procede a cerrar la interfaz de Modificar Funcionalidad, finalizando así el caso de uso.
	<b>8.1.</b> El sistema muestra un mensaje indicando que hay error en la entrada de los datos y vuelve al paso 4.

**Sección “Eliminar Acción”**

Acción del Actor	Respuesta del Sistema
<b>3.</b> El usuario selecciona alguna de las opciones disponibles desde la interfaz 1. (Modificar Funcionalidad o Listar Funcionalidad)	<b>4.</b> El sistema ejecuta el paso 4 del flujo normal de los eventos de las secciones Modificar Funcionalidad o Listar Funcionalidad.
<b>5.</b> El usuario selecciona al usuario que desea eliminar y da clic en la opción Eliminar Funcionalidad.	<b>6.</b> El sistema muestra un mensaje al usuario pidiéndole confirmación de la Funcionalidad que desea realizar. (Ver Interfaz 5)
<b>7.</b> El usuario confirma la Funcionalidad.	<b>8.</b> El sistema elimina la Funcionalidad seleccionada por el usuario y finaliza el caso de uso.

**Interfaz 5**



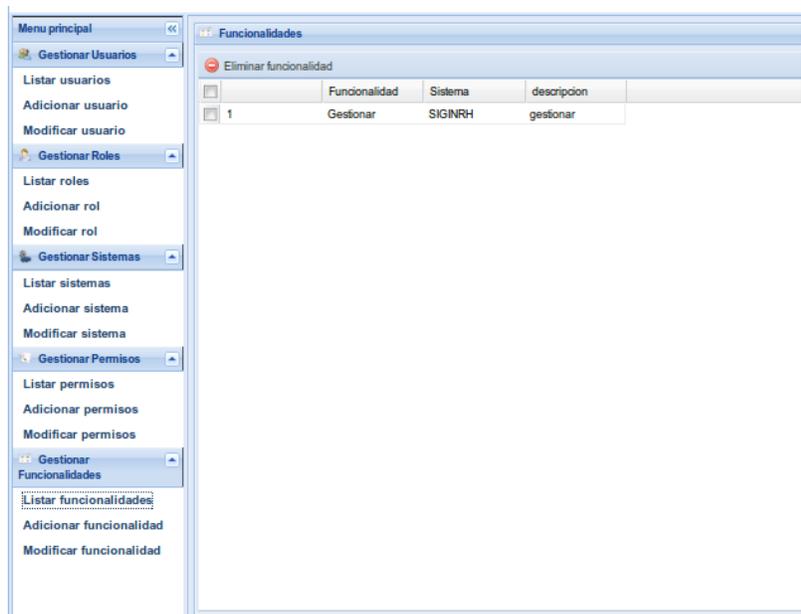
Flujo Alterno.	
Acción del Actor	Respuesta del Sistema
5.1. El usuario no selecciona ningún Funcionalidad y da clic en la opción Eliminar Funcionalidad	6.1 El sistema muestra un mensaje donde se especifica que debe seleccionar una Funcionalidad.
7.1 El usuario no confirma la Funcionalidad.	8.1. El sistema no elimina al Funcionalidad seleccionado por el usuario y finaliza el caso de uso.
<b>Postcondiciones</b>	El sistema inserta, elimina o modifica, las acciones a partir de la opción que se desee realizar.

Tabla 8: Descripción textual del CU Gestionar Funcionalidad

<b>Caso de Uso:</b>	Gestionar Funcionalidades
<b>Actores:</b>	Administrador
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar, eliminar o insertar una funcionalidad al sistema.
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario administrador desea insertar, modificar o eliminar una acción y termina cuando haya sido ejecutada alguna de las funcionalidades anteriormente especificadas.
<b>Precondiciones:</b>	El usuario tiene que estar autenticado.
<b>Referencias</b>	RF 21, RF 22, RF 23, RF 24.
<b>Prioridad</b>	Crítico

<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el usuario desea realizar una de las opciones que se especifican a continuación: (Ver Interfaz 1)</p> <ul style="list-style-type: none"> <li>- Listar Funcionalidades.</li> <li>- Adicionar Acción.</li> <li>- Modificar Acción.</li> <li>- Eliminar Acción.</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Listar Funcionalidades”, ver sección “Listar Funcionalidades”,.”.</li> <li>- Si selecciona “Adicionar Acción”, ver sección “Adicionar Acción”.</li> <li>- Si selecciona “Modificar Acción”, ver sección “Modificar Acción”.</li> <li>- Si selecciona “Eliminar Acción”, ver sección “Eliminar Acción”.</li> </ul>
	<p>3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario.</p>
<b>Prototipo de Interfaz</b>	
<b>Interfaz 1</b>	
<b>Sección “Listar Funcionalidades”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>3. El usuario selecciona la opción Listar Acción (Ver Interfaz 1)</p>	<p>4. El sistema comprueba que existan funcionalidades registrados y procede a mostrar una tabla que contiene el listado de todas las funcionalidades que se encuentran añadidos al sistema. (Ver Interfaz 2)</p>

Interfaz 2



Flujos Alternos

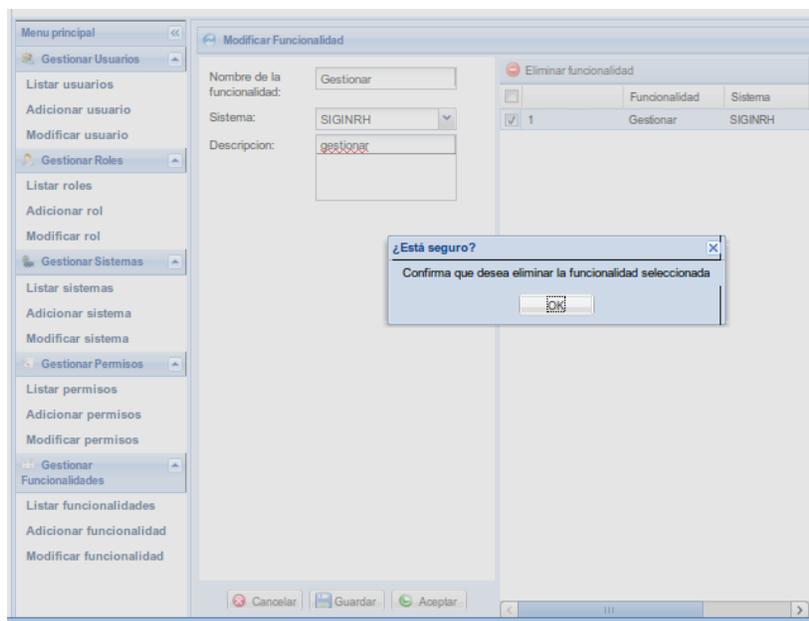
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje de error indicando que no hay funcionalidades añadidas al sistema finalizando así el caso de uso.

Sección “Adicionar Acción”

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona la opción Adicionar Acción. (Ver Interfaz 1)	4. El sistema muestra un formulario que le permite al usuario insertar los datos (Acción y Descripción).
5. El usuario inserta los datos (Acción y	6. El sistema valida que todos los campos estén

Descripción).y da clic en el botón guardar.	correctos.
	7. El sistema verifica que la acción no se encuentre insertada.
	8. El sistema guarda la nueva acción manteniendo la interfaz de Adicionar Acción finalizando así el caso de uso.

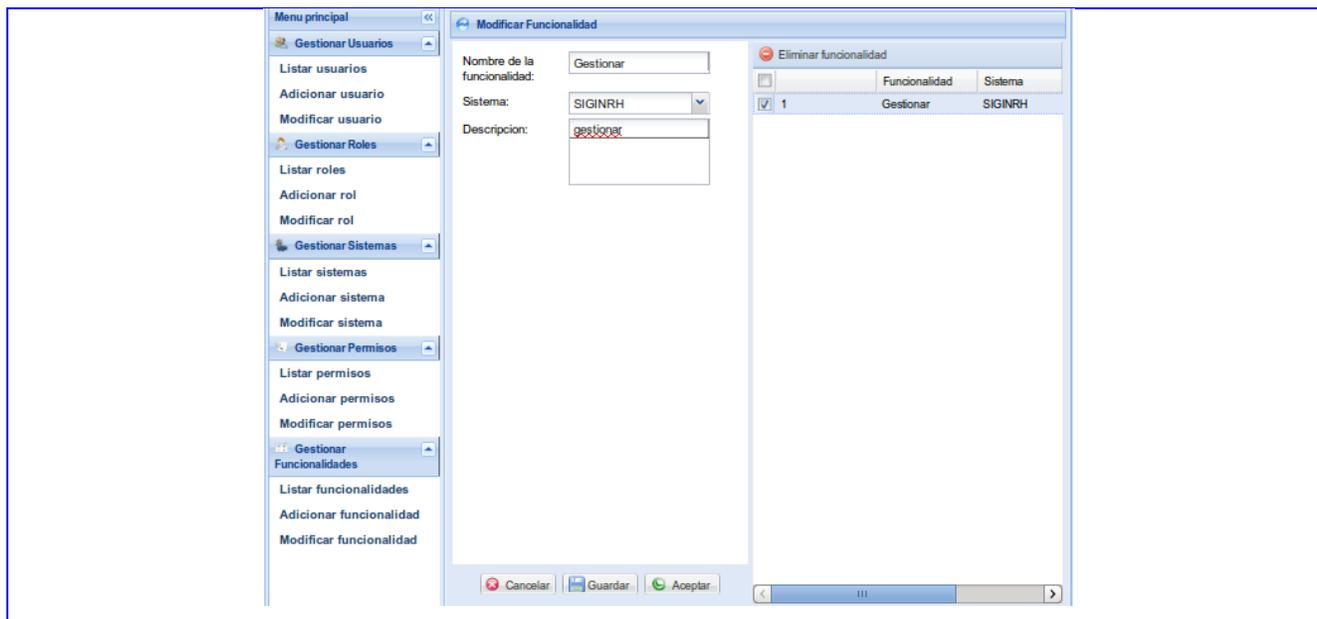
**Interfaz 3**



**Flujo Alterno**

Acción del Actor	Respuesta del Sistema
5.1 El usuario inserta los datos (Acción y Descripción). y da clic en el botón Aceptar.	8.1 Realiza los pasos 6 y 7 del flujo normal de los eventos, guarda la acción y procede a cerrar la interfaz de Adicionar Acción, finalizando así el caso de uso.

	<p><b>6.1</b> El sistema le informa al usuario que hay error en la entrada de los datos y el usuario vuelve al paso 3 del flujo normal de los eventos.</p>
	<p><b>7.1</b> El sistema envía un mensaje de error especificando que el usuario se encuentra registrado en el sistema, y vuelve al paso 3 si desea adicionar otra acción, en caso contrario finaliza el caso de uso.</p>
<p><b>Sección “Modificar Acción.”</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p><b>3.</b> El usuario selecciona la opción Modificar Acción (Ver Interfaz 1)</p>	<p><b>4.</b> Ir al paso 2 del flujo normal de los eventos de la sección Listar Funcionalidades.</p>
<p><b>5.</b> El usuario selecciona la acción que desea modificar. (Ver Interfaz 4)</p>	<p><b>6.</b> El sistema llena el formulario con los datos que tiene guardado del usuario seleccionado.</p>
<p><b>7.</b> El usuario modifica los datos (Acción y Descripción) y da clic en el botón guardar.</p>	<p><b>8.</b> El sistema valida que todos los campos estén correctos.</p>
	<p><b>9.</b> El sistema guarda los cambios realizados por el usuario, manteniendo la interfaz de Modificar Acción, finalizando así el caso de uso.</p>
<p><b>Interfaz 4</b></p>	



**Flujo Alterno**

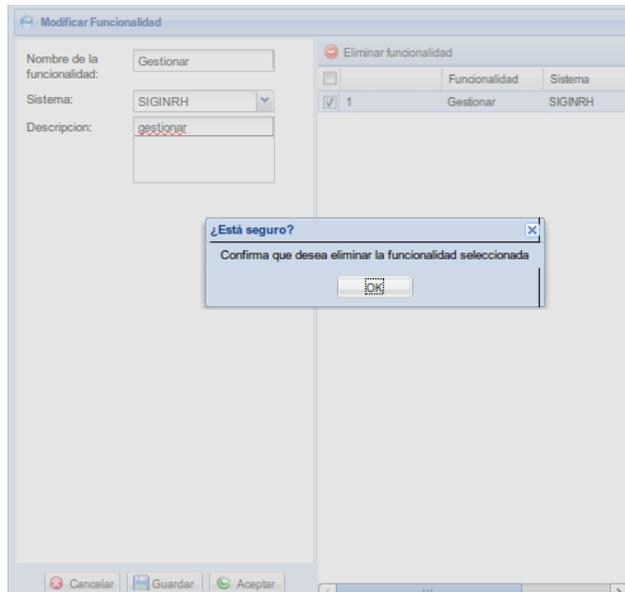
Acción del Actor	Respuesta del Sistema
7.1. El usuario modifica los datos (Acción y Descripción) y da clic en el botón aceptar.	9.1. El sistema realiza el paso 8 del flujo normal de eventos y procede a cerrar la interfaz de Modificar Acción, finalizando así el caso de uso.
	8.1. El sistema muestra un mensaje indicando que hay error en la entrada de los datos y vuelve al paso 4.

**Sección “Eliminar Acción”**

Acción del Actor	Respuesta del Sistema
3. El usuario selecciona alguna de las opciones disponibles desde la interfaz 1. (Modificar Acción o Listar Acción)	4. El sistema ejecuta el paso 4 del flujo normal de los eventos de las secciones Modificar Acción o Listar Acción.
5. El usuario selecciona al usuario que desea eliminar y da clic en la opción Eliminar Acción.	6. El sistema muestra un mensaje al usuario pidiéndole confirmación de la acción que desea realizar.

7. El usuario confirma la acción.	8. El sistema elimina la acción seleccionada por el usuario y finaliza el caso de uso.
-----------------------------------	--

**Interfaz 5**



**Flujo Alternativo.**

Acción del Actor	Respuesta del Sistema
5.1. El usuario no selecciona ningún acción y da clic en la opción Eliminar Acción	6.1 El sistema muestra un mensaje donde se especifica que debe seleccionar una funcionalidad..
7.1 El usuario no confirma la acción.	8.1. El sistema no elimina al acción seleccionado por el usuario y finaliza el caso de uso.
<b>Postcondiciones</b>	El sistema inserta, elimina o modifica, las funcionalidades a partir de la opción que se desee realizar.

Tabla 9: Descripción textual de CU Gestionar Recurso Hidráulico.

<b>Caso de Uso:</b>	Gestionar Recurso Hidráulico	
<b>Actores:</b>	Administrador	
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de modificar o listar los recursos hidráulicos	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario administrador desea modificar, o listar recurso hidráulico y termina cuando haya sido ejecutada alguna de las acciones anteriormente especificadas.	
<b>Precondiciones:</b>	El usuario tiene que estar autenticado.	
<b>Referencias</b>	RF 27, RF 28.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El caso de uso se inicia cuando el usuario desea realizar una de las opciones que se especifican a continuación: (Ver Interfaz 1)</p> <ul style="list-style-type: none"> <li>- Listar Recurso Hidráulico.</li> <li>- Modificar Recurso Hidráulico.</li> </ul>	<p>2. El sistema realiza la operación según la opción seleccionada por el usuario.</p> <ul style="list-style-type: none"> <li>- Si selecciona “Listar Recurso Hidráulico”, ver sección “Listar Recurso Hidráulico”.</li> <li>- Si selecciona “Modificar Recurso Hidráulico”, ver sección “Modificar Recurso Hidráulico”.</li> </ul>	
	<p>3. El caso de uso termina cuando el sistema procesa la información según la acción realizada por el usuario.</p>	
<b>Sección “Listar Recurso Hidráulico”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	

	3. El sistema comprueba que existan Recurso Hidráulico registrados y procede a mostrar una tabla que contiene el listado de todos los Recurso Hidráulico que se encuentran añadidos al sistema. (Ver Interfaz 2)
<b>Sección “Modificar Recurso Hidráulico.”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3. El usuario selecciona la opción Modificar Recurso Hidráulico (Ver Interfaz 1)	4. Ir al paso 2 del flujo normal de los eventos de la sección Listar Recurso Hidráulico.
5. El usuario selecciona el Recurso Hidráulico que desea modificar. (Ver Interfaz 4)	6. El sistema llena el formulario con los datos que tiene guardado del Recurso Hidráulico seleccionado.
7. El usuario modifica los datos (Capacidad y Estado) y da clic en el botón guardar.	8. El sistema valida que todos los campos estén correctos.
	9. El sistema guarda los cambios realizados por el usuario, manteniendo la interfaz de Modificar Recurso Hidráulico, finalizando así el caso de uso.
<b>Interfaz 4</b>	
<b>Flujo Alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	8.1. El sistema muestra un mensaje indicando que hay error en la entrada de los datos y vuelve al paso 4.
<b>Postcondiciones</b>	El sistema lista o modifica, los Recursos Hidráulicos a partir de la opción que se desee realizar.

<b>Caso de Uso:</b>	Autenticar Usuario.	
<b>Actores:</b>	Administrador	
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de que los usuarios se puedan autenticar para poder interactuar con el sistema.	
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario registrado en el sistema intenta acceder a las funcionalidades del sistema y este le pide que se identifique como tal.	
<b>Precondiciones:</b>	El usuario tiene que estar insertado en el sistema.	
<b>Referencias</b>	RF 25.	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El caso de uso se inicia cuando el usuario intenta acceder al sistema.	2. El sistema realiza la operación de mostrarle la ventana de login al usuario.	
3. El usuario inserta su nombre de usuario y contraseña.	4. El sistema valida los campos.	
	5. El sistema busca si ese usuario pertenece y si su contraseña se corresponde con el usuario que intenta acceder. Muestra la interfaz de acuerdo a los permisos que tenga asignados el usuario y a las funcionalidades que pueda acceder.	
<b>Prototipo de Interfaz</b>		

<i>Interfaz 1</i>	
	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p><b>4.1.</b> El sistema muestra un mensaje al usuario indicando que los campos son obligatorios y se repite paso 3.</p>
	<p><b>3.</b> El sistema muestra un mensaje indicando que la contraseña no se corresponde con el nombre de usuario insertado y vuelve al paso 3.</p>
Postcondiciones	Se logra que un usuario se identifique en el sistema.

Tabla 10: Descripción textual de CU Autenticar Usuario.

Anexo 2 Vista lógica de la Arquitectura

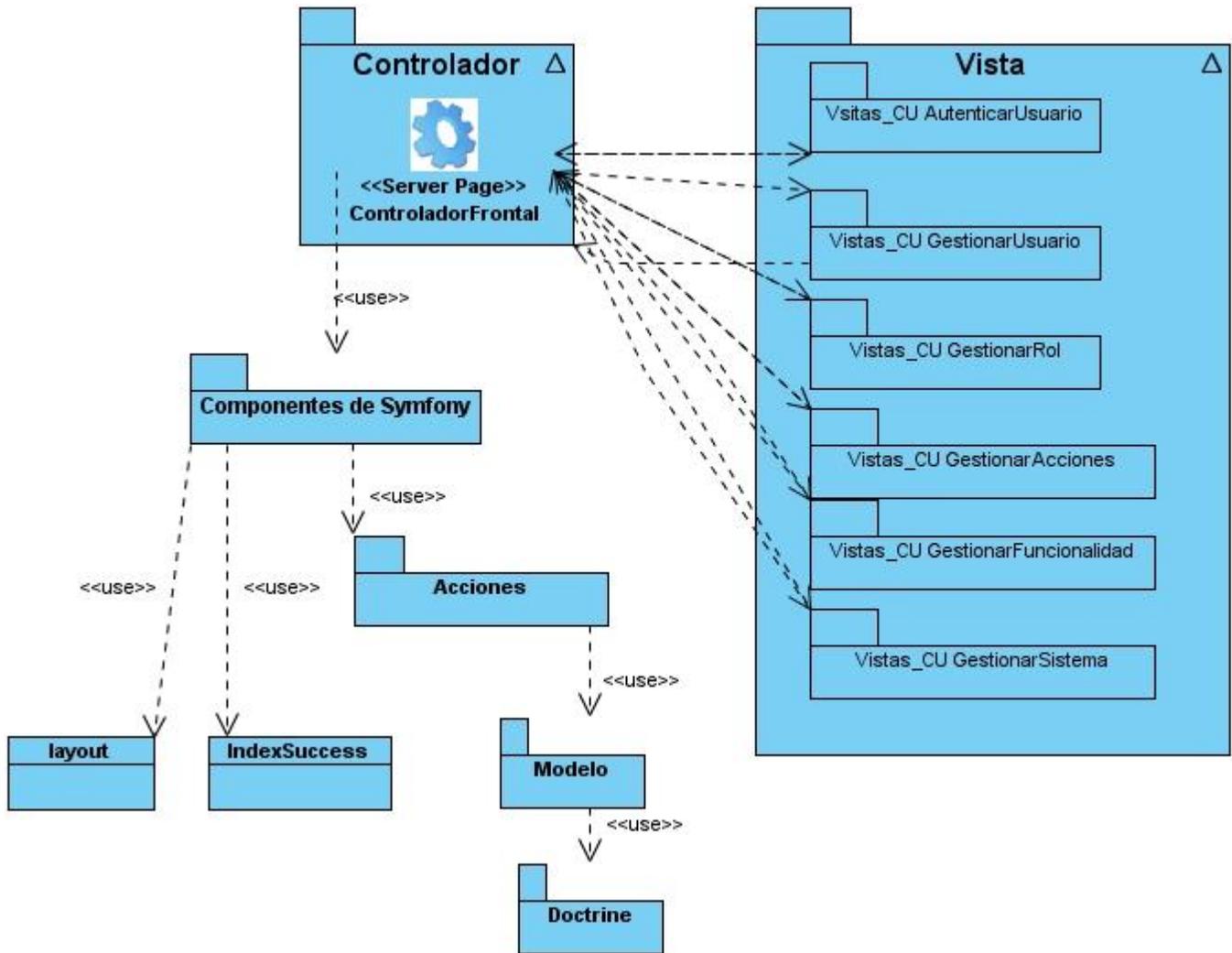


Ilustración 4: Vista lógica de la Arquitectura

Anexo 3 Diagrama de Clases del Diseño.

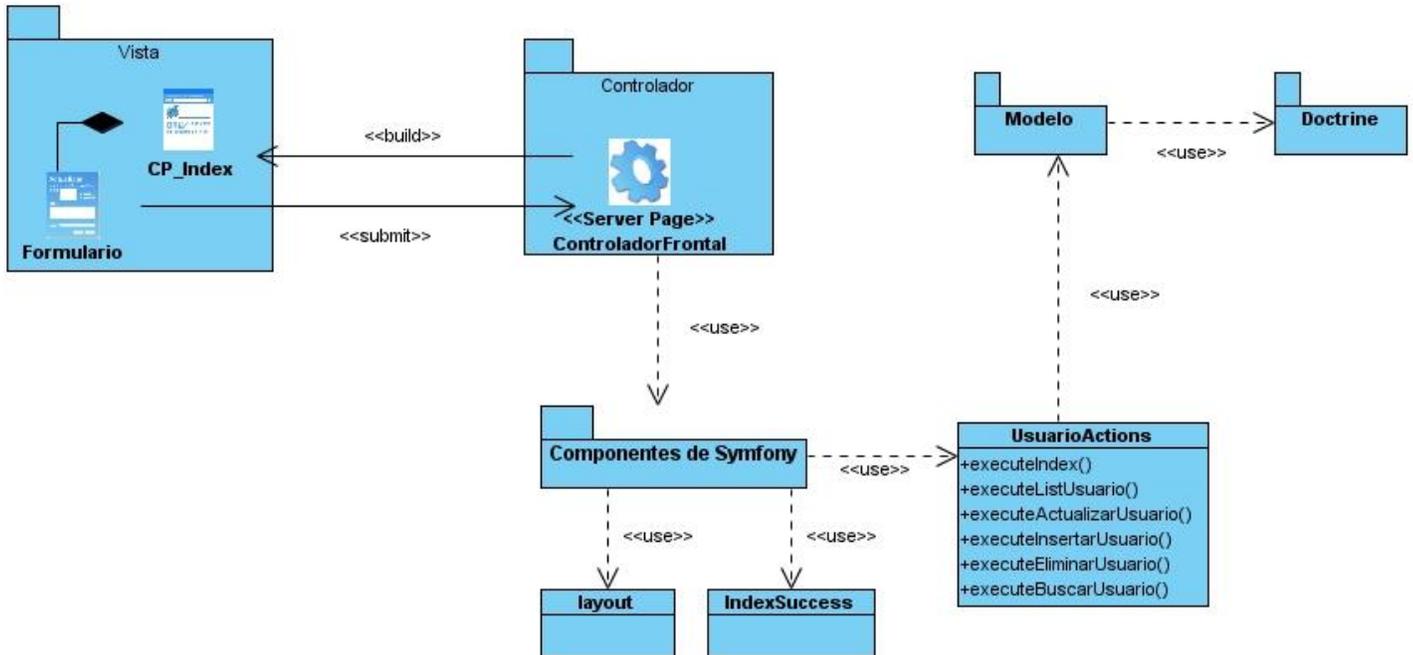


Ilustración 5: DCD Autenticar Usuario.

Ilustración 6: DCD Gestionar Usuario

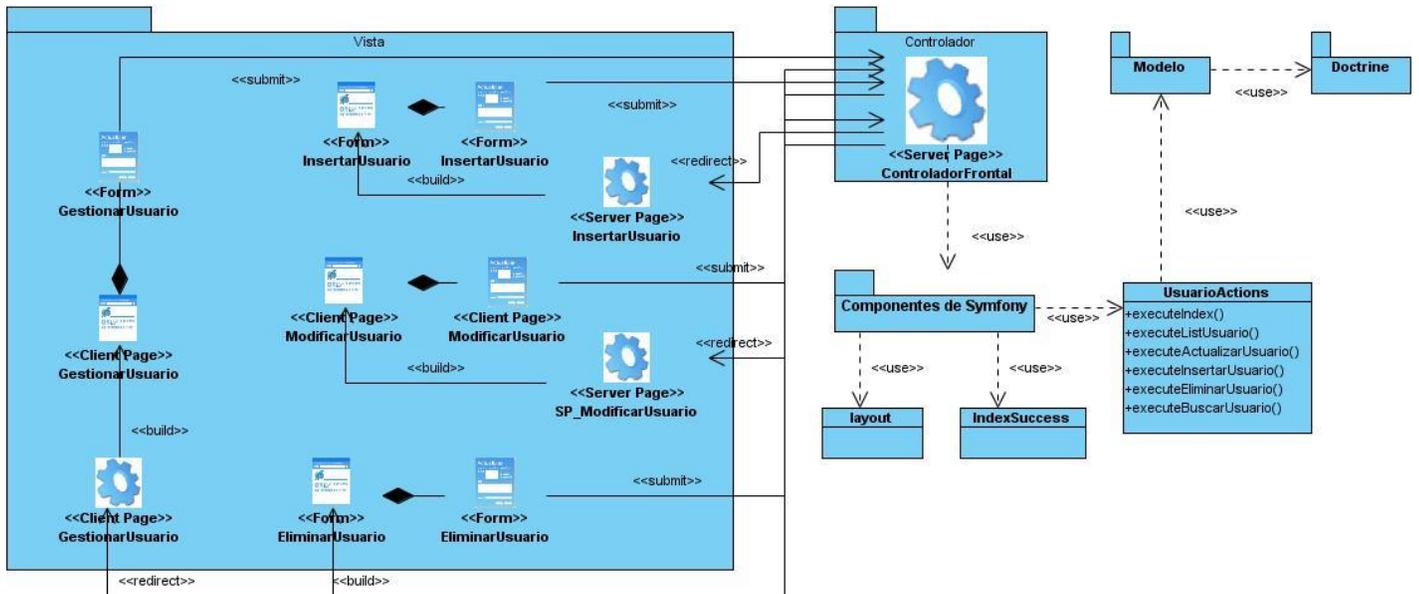


Ilustración 7: DCD Gestionar Roles

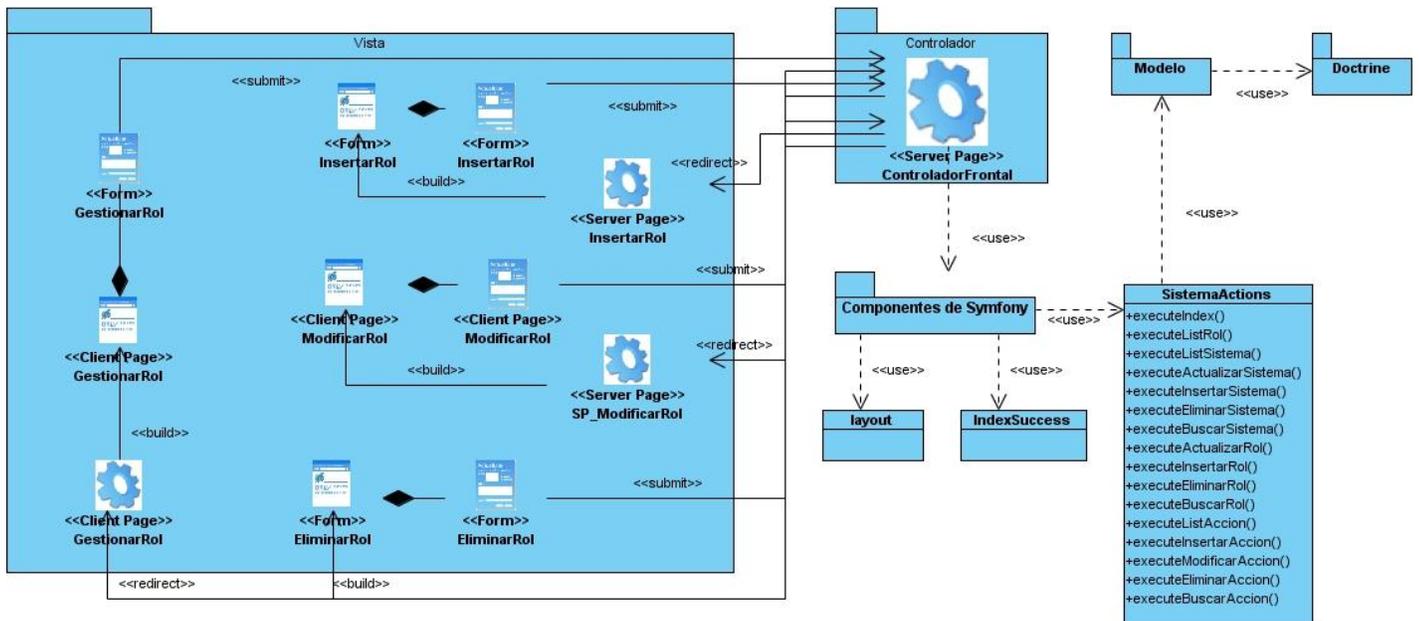


Ilustración 8: DCD Gestionar Sistema

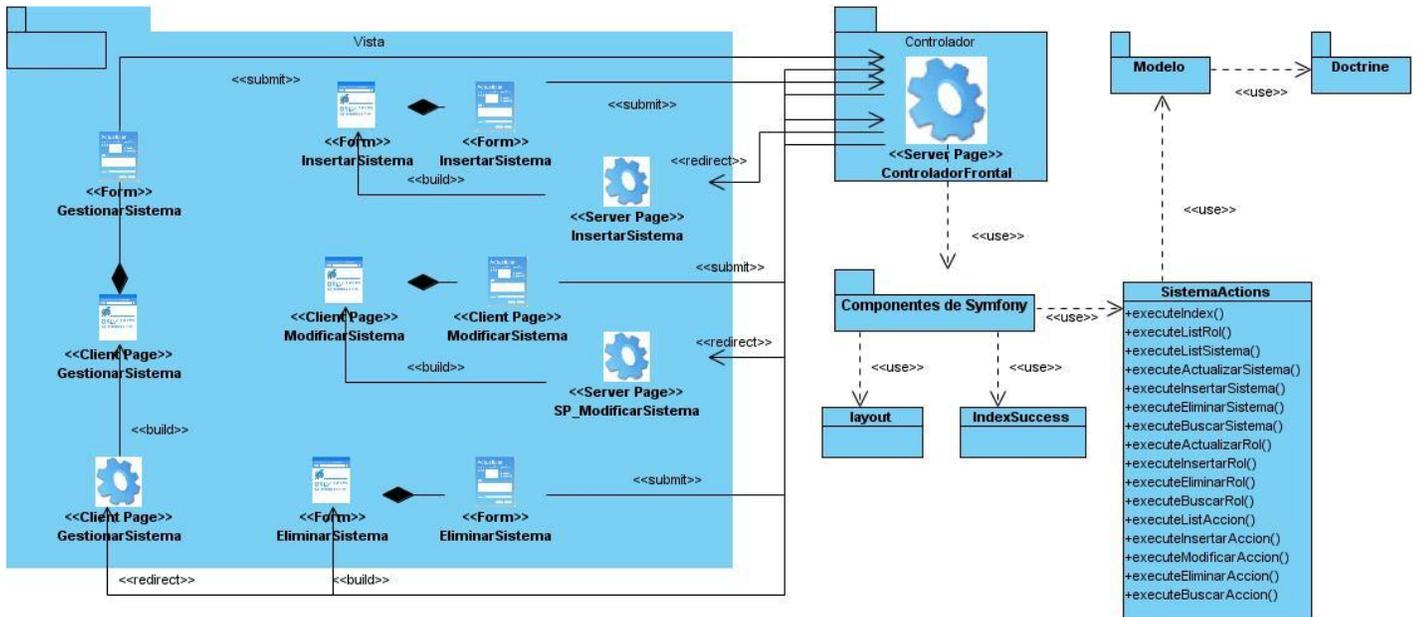


Ilustración 9: DCD Gestionar Funcionalidad

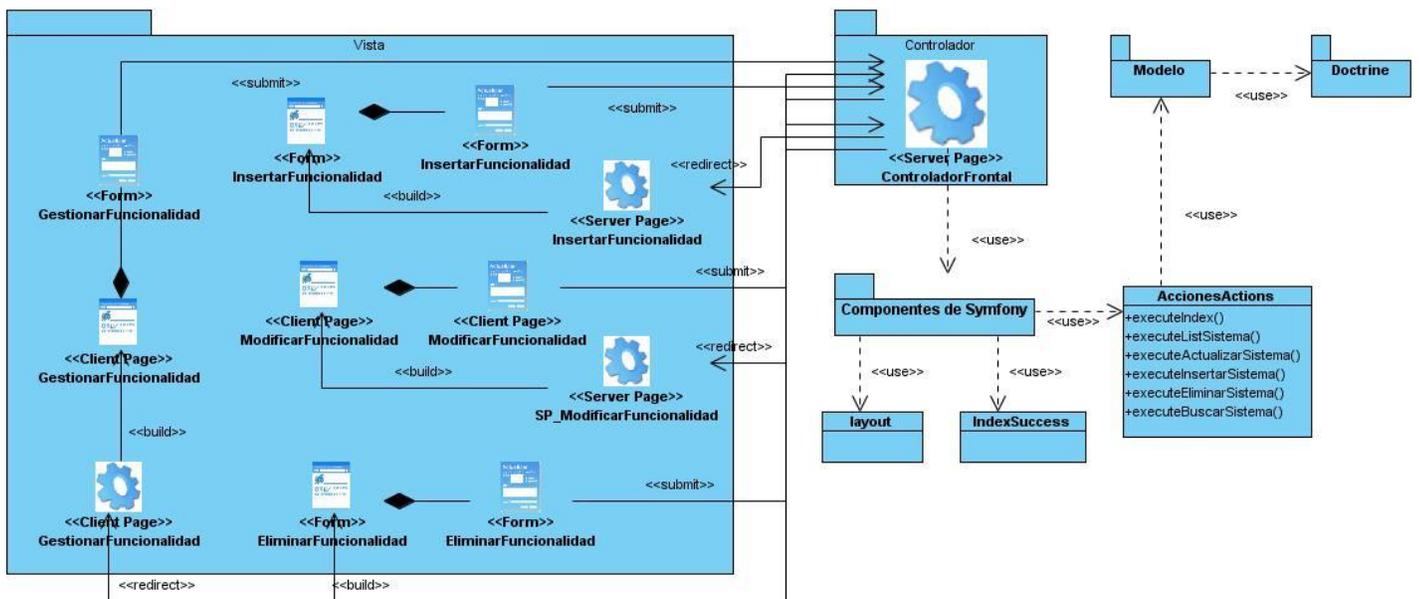


Ilustración 10: DCD Gestionar Acciones

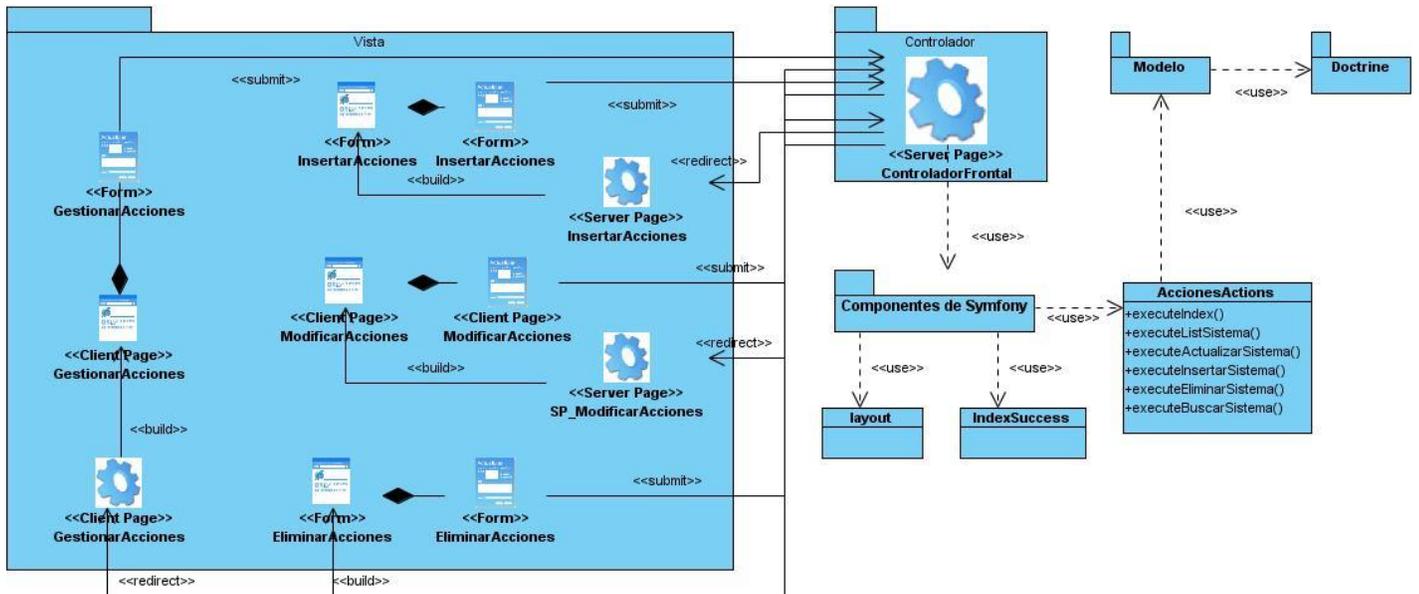
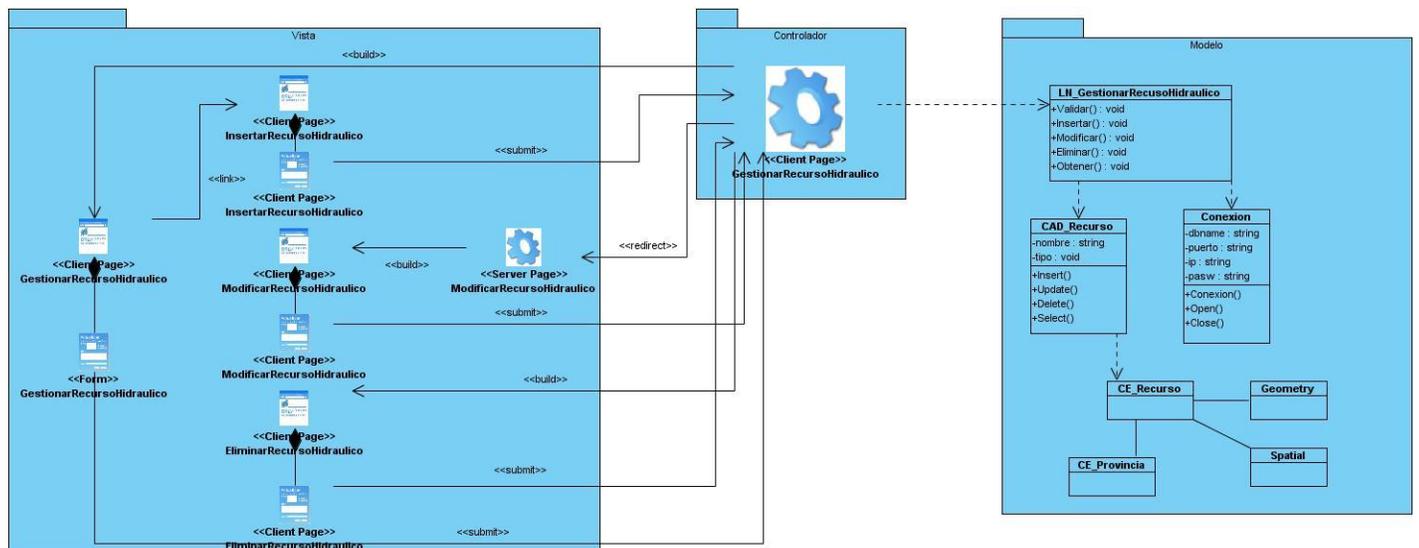


Ilustración 11: DCD Gestionar Recurso Hidráulico.



Anexo 4 Diseño de Bases de Datos.

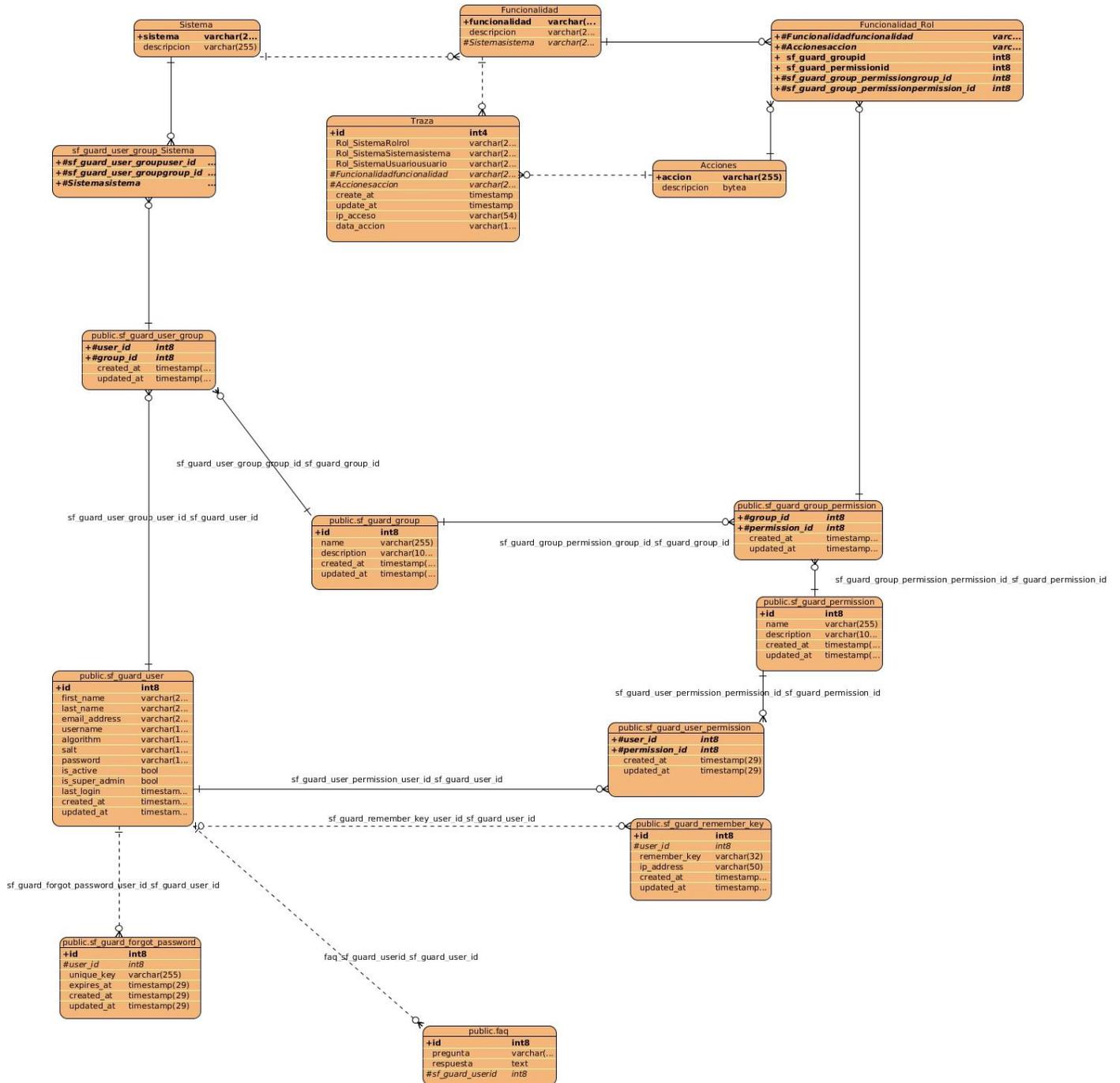


Ilustración 12: DER Módulo de Administración.

Ilustración 13: DER Módulo de Gestión de la Información

geometry_columns		
<b>+f_table_catalog</b>	<b>varchar(256)</b>	<b>Nullable = false</b>
<b>+f_table_schema</b>	<b>varchar(256)</b>	<b>Nullable = false</b>
<b>+f_table_name</b>	<b>varchar(256)</b>	<b>Nullable = false</b>
<b>+f_geometry_column</b>	<b>varchar(256)</b>	<b>Nullable = false</b>
coord_dimension	int4	Nullable = false
srid	int4	Nullable = false
type	varchar(30)	Nullable = false

spatial_ref_sys		
<b>+srid</b>	<b>int4</b>	<b>Nullable = false</b>
auth_name	varchar(256)	Nullable = true
auth_srid	int4	Nullable = true
srsrtext	varchar(2048)	Nullable = true
proj4text	varchar(2048)	Nullable = true

embalses		
<b>+gid</b>	<b>int4</b>	<b>Nullable = false</b>
tipo	varchar(20)	Nullable = true
nombre	varchar(100)	Nullable = true
categoria	varchar(15)	Nullable = true
the_geom	int4	Nullable = true

provincias		
<b>+gid</b>	<b>int4</b>	<b>Nullable = false</b>
TIPOELEMEN	varchar(10)	Nullable = true
NOMBRE	varchar(20)	Nullable = true
the_geom	int4	Nullable = true

ws_consulta

Anexo 5 Diagrama de Componentes.

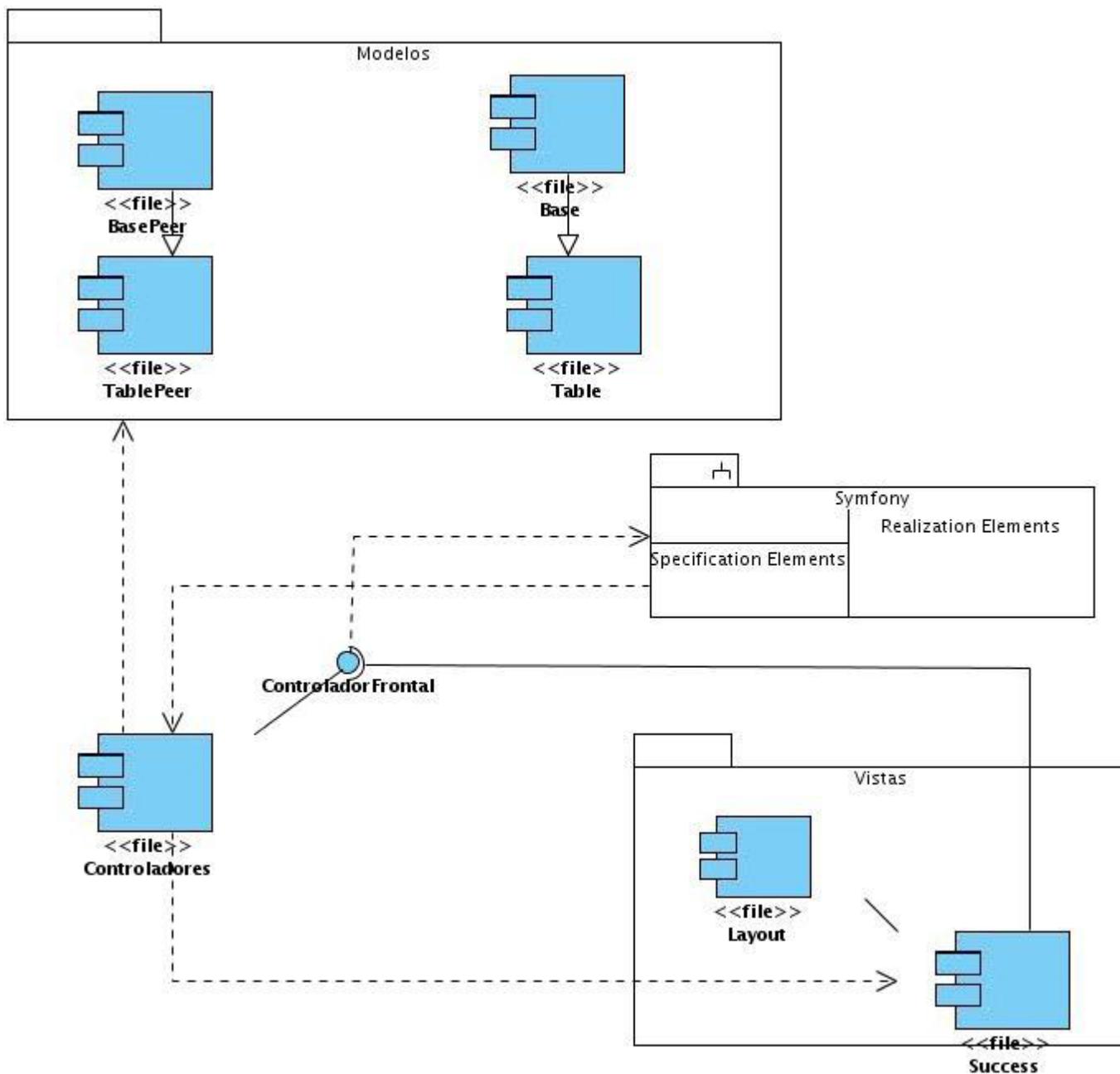


Ilustración 14: Diagrama de Componentes

## GLOSARIO DE TÉRMINOS

### A

- **Autenticación.**

Es el proceso de descubrir y comprobar la identidad de un usuario mediante el examen de sus credenciales y la validación de las mismas consultando a una autoridad determinada. Actualmente se utilizan una gran variedad de mecanismos de autenticación.

- **Autorización.**

Es el proceso que determina si el usuario o proceso previamente identificado y autenticado tiene permitido el acceso a los recursos.

- **AJAX.**

Acónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página

- **API.**

Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

### C

- **Control de Acceso.**

Es el proceso de conceder permisos a usuarios de acceder a objetos, información, datos como ficheros o impresoras en la red.

- **Confidencialidad.**

La información o los activos informáticos son accedidos solo por las personas autorizadas. Sería fácil mantener la confidencialidad de un sistema si nadie tuviera acceso a él, pero este tendría una disponibilidad nula. (Sotelo 2008)

**D**

- **Disponibilidad.**

La información solo puede ser accedida por las personas autorizadas en el momento requerido.  
(Sotelo 2008)

**F**

- **Framework.**

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

**H**

- **Herramienta CASE.**

(Computer Aided/Assisted Software/System Engineering) o (Ingeniería de Software Asistida por Computadora) son herramientas utilizadas en el modelamiento visual de sistemas.

**I**

- **INRH**

Es la institución que se encarga de la administración y control de todos los objetivos hidráulicos cubanos.

- **Integridad**

La información que está en correspondencia con la realidad solo puede ser modificada por las personas autorizadas y de la forma autorizada.

- **IDE**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

**R**

- **RUP.**

El Proceso Unificado Racional o de Desarrollo (Rational Unified Process en inglés, tradicionalmente abreviado como RUP) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

**S**

- **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.

- **SIG**

Sistema de Información Geográfica (GIS en Inglés).

- **SIGINRH**

Sistema de Información Geográfica del Instituto Nacional de Recursos Hidráulicos.

**U**

- **UML**

Un lenguaje para modelar objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos.