

**Universidad de las Ciencias Informáticas**

**Facultad 15**



# **Propuesta de Herramienta para la Gestión del Proceso de Evaluación de Software de los Centros de Producción**

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

**Autor(es): José Otilio Díaz Esquivel**

**Loreta Naybi Hernández Fernández**

**Tutor Ing. Geiser Arcio Pérez Rivas**

Ciudad Habana, junio del 2010

## Declaración de Autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

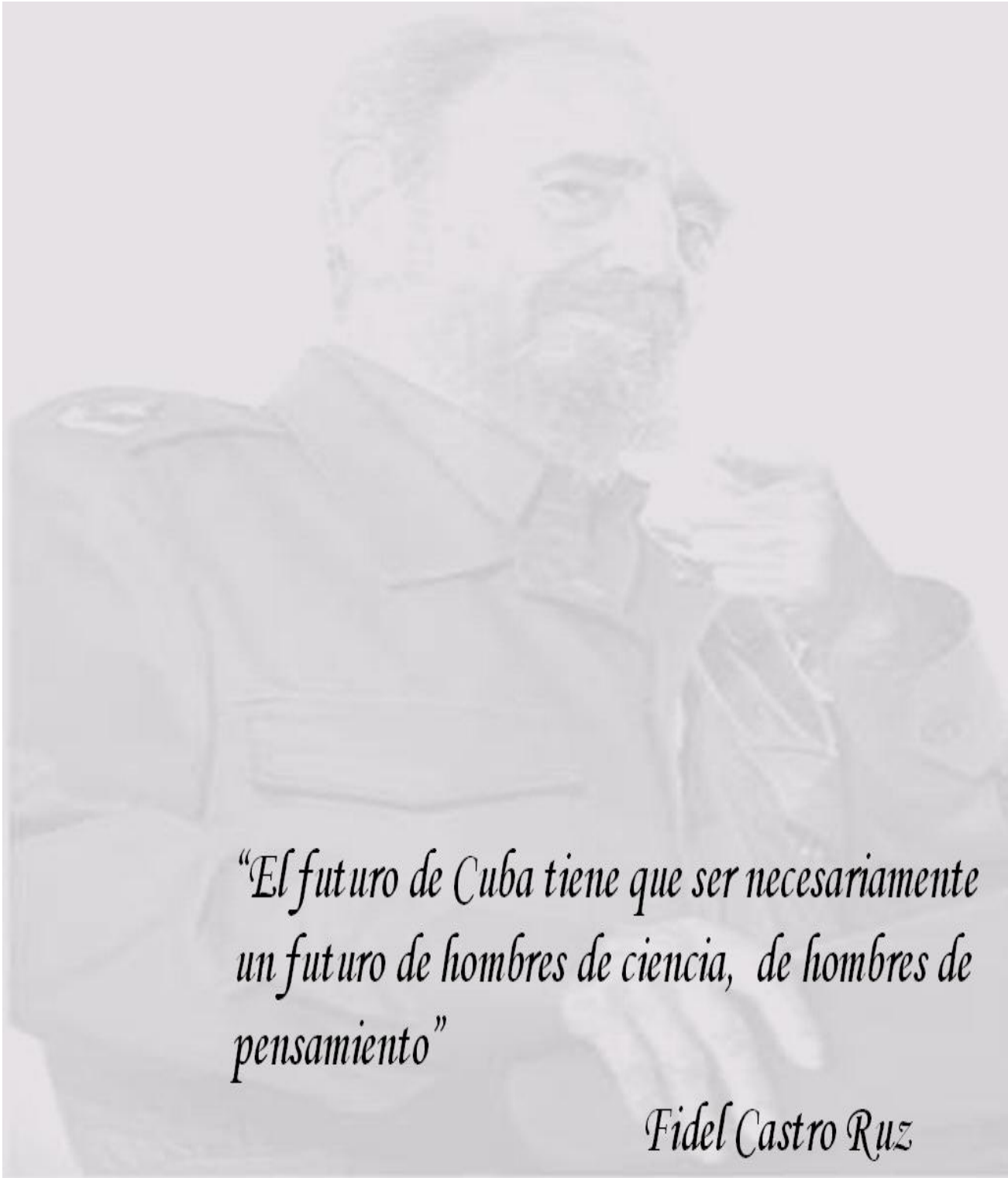
Para que así conste firmo la presente a los \_\_\_\_ días del mes de junio del año \_\_\_\_\_.

---

Firma del Autor

---

Firma del Tutor



*“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento”*

*Fidel Castro Ruz*

*Agradezco a todas aquellas personas que durante estos 5 Años me apoyaron y me guiaron en mis estudios, a mis compañeros de aula a mis amigos, ellos que fueron mi consuelo y mi empuje cuando las cosas se tronaban difíciles.*

*Muy especialmente a mi madre por ser mi principal estímulo durante toda mi vida estudiantil y personal.*

*A mi familia por ser mi empuje y máximo compromiso*

*A Pepe por su empeño y dedicación.*

*A mi pareja por su paciencia y apoyo durante todo este tiempo. A su familia porque siempre creyó en mí. Gracias por estar a mi lado.*

*Loreta*

*A mis padres por todo el sacrificio que hicieron para que este sueño se hiciera realidad, por su paciencia dedicación, amor y apoyo todo este tiempo que he estado lejos. A ellos les estaré eternamente agradecido pues les debo lo que soy.*

*A mi familia por ofrecerme tanto amor y cariño desde pequeño, por todos los momentos buenos y malos que hemos vivido y por ayudarme a ser el hombre que soy.*

*A mis compañeros de aula y amigos, los de aquí y los de allá, por darme la oportunidad de conocerlos, por su apoyo incondicional, por ser tan especiales y estar ahí siempre que los necesité en los buenos y en los malos momentos.*

*A mis entrenadores y compañeros del equipo de Béisbol por estos 5 años compartiendo como hermanos.*

*A mi compañera de Tesis, por todo su esfuerzo y por confiar en mí.*

*A nuestro Tutor, por todo su empeño y dedicación por lograr que este trabajo saliera adelante. A la Revolución y a Fidel por darme la oportunidad de ser parte de un proyecto como este.*

*Pepe*

*Dedico este trabajo de diploma a mi Abuela que aunque ya no está entre nosotros siempre guió mi educación para que un día como hoy; yo obtuviera estos resultados.*

*También dedico este trabajo a mi madre, a mis hermanas, mi padre de formación (Mi abuelito), y a mi padre de Vida, él junto a mi madre son los máximos responsables de que mi título se vea materializado.*

*Loreta*

*Dedico este trabajo a mis padres, por toda la confianza que pusieron en mí y por darme la fuerza para seguir adelante. A mi hermanita que es mi vida, a mis dos sobrinitos que trajeron tanta alegría a la familia. A Yami, Edgar, Elier, Edgarito, Mima, Ángel, Tai, Pepe, Cabin, Ana, Rubén, Manolo, Elisa, Mayi, Luisi, en fin a toda mi familia, amigos y vecinos, por darme su apoyo incondicional en todo momento.*

*Especialmente quiero dedicar este trabajo a mi abuelita Amalia que la vida no le dio la oportunidad de ver este momento pero sé que estaría orgullosa de mí.*

*Pepe*

La Universidad de las Ciencias Informáticas (UCI) juega un papel importante en la producción de software, pretende convertirse en una universidad innovadora de excelencia científica y productiva, distinguida por el uso de las tecnologías. El reto que se impone es grande, esto hace que se deban tomar decisiones acertadas de acuerdo a la calidad de los productos que en ella se generan.

El Departamento de Pruebas es el área encargada de gestionar todo el proceso por el que debe transitar un producto software para su liberación. En términos de liberación este centro, es el máximo responsable de que los productos que en la UCI se generan salgan a su explotación con la más óptima calidad, así como que toda la documentación, dígame estudios y artefactos generados durante el desarrollo del producto, tengan un nivel máximo.

En su funcionamiento el Departamento cuenta con varias ineficiencias como que no se establece una guía predeterminada de requisitos para la evaluación, actualmente no se cuenta con un patrón de pruebas que permita determinar una evaluación cualitativa y cuantitativa de un producto, las pruebas son hechas sin un enfoque objetivo de lo que se va a evaluar, además de que la gama de pruebas existentes es pequeña sin contar que no existe una herramienta que tribute a su realización, finalmente se recolectan los datos pero no tienen concordancia con las métricas existentes y no se hace una retroalimentación con toda la información. Por estas razones en el Departamento se planteó la necesidad de un proceso más organizado, eficiente y detallado para un mejor servicio por parte del mismo, con el fin que la información sea fácilmente accesible por todos los interesados en el tema de la calidad y que se logre un proceso de evaluación más confiable.

Por lo que se hizo necesaria una herramienta que, realizadas las pruebas a cada producto sea capaz de automatizar todos los datos recolectados en este proceso, dejándolos reflejados de una forma más organizada, permitiendo establecer consultas, además de ejecutar operaciones con estos datos, como el cálculo de la calidad del producto, posibilitando establecer comparaciones futuras entre que tan mejor o peor es este producto con respecto a otros, para darle respuesta a esta necesidad se desarrolló una aplicación, que brinda como servicio principal calcular la calidad de un producto software, fácilmente navegable por los usuarios del sistema y que se espera cumpla con todas las expectativas creadas por parte del Departamento de Pruebas.

<b>Introducción</b> .....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN DEL TEMA</b> .....	5
<b>Introducción</b> .....	5
<b>Aplicaciones Web</b> .....	5
<b>Ventajas del uso de aplicaciones Web</b> .....	5
<b>Tipos de aplicaciones Web</b> .....	6
<b>Análisis de Otros Portales Existentes</b> .....	8
<b>Aspectos importantes para crear una Web para Evaluar la Calidad de un Producto Software</b> .....	10
<b>Tecnologías y Herramientas Propuestas para el Desarrollo del Sistema</b> .....	10
<b>Metodología de desarrollo</b> .....	10
<b>Metodología Ágil XP (Extreme Programming)</b> .....	11
<b>Metodología de Desarrollo Rational Unified Process (RUP)</b> .....	11
<b>Justificación de la Metodología Seleccionada</b> .....	11
<b>Herramientas para la Modelación Visual</b> .....	12
<b>Visual Paradigm</b> .....	12
<b>Rational Rose</b> .....	13
<b>Justificación de la Herramienta para la Modelación Visual</b> .....	13
<b>Lenguaje de Representación Visual para el Modelado del Sistema</b> .....	13
<b>Lenguajes de Programación y Herramientas para la Web</b> .....	14
<b>PHP</b> .....	14
<b>Java</b> .....	15
<b>Groovy and Grails</b> .....	16
<b>Justificación de la Lenguajes de Programación y Herramientas para la Web</b> .....	16
<b>Sistema de Base de Datos</b> .....	17

<b>Propuesta de Solución</b> .....	17
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	19
<b>Introducción</b> .....	19
<b>Problema</b> .....	19
<b>Objeto de Automatización</b> .....	19
<b>Información que se Maneja</b> .....	20
<b>Modelo de Dominio</b> .....	20
<b>Justificación de la utilización del Modelo de Dominio</b> .....	20
<b>Descripción del Modelo de Dominio</b> .....	20
<b>Reglas</b> .....	21
<b>Actores y Trabajadores</b> .....	21
<b>Diagrama de clases del Modelo de Dominio</b> .....	22
<b>Modelo de Objetos</b> .....	23
<b>Especificación de los Requisitos de Software</b> .....	24
<b>Requerimientos No Funcionales</b> .....	28
<b>Definición de los Casos de Uso del Sistema</b> .....	29
<b>Actores del Sistema</b> .....	29
<b>Diagrama de Caso de Uso del Sistema</b> .....	29
<b>Descripción de los Casos de Uso del Sistema</b> .....	30
<b>Conclusiones Parciales</b> .....	49
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA</b> .....	50
<b>Introducción</b> .....	50
<b>Modelado de Análisis</b> .....	50
<b>Diagrama de clases del análisis</b> .....	50



<b>Entidades de clases de análisis .....</b>	<b>51</b>
<b>Arquitectura de Software .....</b>	<b>51</b>
<b>Descripción de la Arquitectura Utilizada.....</b>	<b>52</b>
<b>Definiciones de diseño que se aplican .....</b>	<b>52</b>
<b>Diseño .....</b>	<b>53</b>
<b>Diagrama de interacción.....</b>	<b>53</b>
<b>Diagrama de Secuencia del diseño .....</b>	<b>54</b>
<b>Diagrama de clases del diseño .....</b>	<b>54</b>
<b>Diseño de Base de Datos.....</b>	<b>54</b>
<b>Modelo de datos .....</b>	<b>55</b>
<b>Conclusiones Parciales.....</b>	<b>56</b>
<b>CAPÍTULO 4: Implementación.....</b>	<b>57</b>
<b>Introducción.....</b>	<b>57</b>
<b>Mapa de Navegación.....</b>	<b>57</b>
<b>Módulos .....</b>	<b>58</b>
<b>Modelo de despliegue.....</b>	<b>59</b>
<b>Conclusiones Parciales.....</b>	<b>60</b>
<b>Conclusiones Generales .....</b>	<b>61</b>
<b>Recomendaciones.....</b>	<b>62</b>
<b>Bibliografía.....</b>	<b>63</b>
<b>Bibliografía Referenciada .....</b>	<b>64</b>
<b>Anexo 1: Diagramas de clases de Análisis .....</b>	<b>65</b>
<b>Anexo 2: Diagramas de secuencia del diseño.....</b>	<b>65</b>
<b>Anexo 3: Diagrama de Clases de Diseño Web.....</b>	<b>69</b>

### **INTRODUCCIÓN**

---

El uso irremplazable de las Tecnologías de la Informática y las Comunicaciones (TIC) se está haciendo más evidente en las últimas décadas. El desarrollo de las mismas permite que los procesos de la humanidad se realicen con más calidad, control y velocidad, siendo posible acortar innumerables situaciones complicadas en la sociedad y aumentar la fiabilidad y efectividad de los bienes y servicios que brindan en la actualidad el uso de las TIC.

La fuerte evolución tecnológica actual provoca que la sociedad demande de la informática la capacidad de desarrollar un gran número de aplicaciones cada vez más grandes y en unos plazos de tiempo cada vez más cortos. Esto tiene como consecuencia que el software desarrollado no siempre tenga la calidad necesaria. Esta falta de calidad se manifiesta en posteriores fallos y defectos que generan pérdidas importantes y altos costes de mantenimiento.

El principal obstáculo para garantizar el aseguramiento de la calidad es la necesidad de dedicar una cantidad de recursos humanos y un tiempo de los que normalmente no se dispone; por ello se plantea la necesidad de desarrollar herramientas automáticas que ayuden en esta tarea. Teniendo en cuenta que garantizar la calidad total del software es actualmente un objetivo prácticamente inalcanzable, un objetivo viable es construir herramientas que ayuden a garantizar una calidad aceptable a un costo razonable.

Para lograr el objetivo de informatizar la sociedad cubana, en la UCI se crean gran número de proyectos, los que recogen, estudios y desarrollo de aplicaciones en alguna rama de la economía o la sociedad. El Departamento de Pruebas es el área encargada en la UCI de comprobar que los productos de software estén desarrollados con la máxima calidad.

En el funcionamiento del Departamento existen una serie de problemas como que no se establece una guía predeterminada de requisitos para la evaluación, actualmente no se cuenta con un patrón de pruebas que permita determinar una evaluación cualitativa y cuantitativa de un producto, las pruebas son hechas sin un enfoque objetivo de lo que se va a evaluar, además de que la gama de pruebas existentes es pequeña sin contar que no existe una herramienta que tribute a su realización, finalmente se recolectan los datos pero no tienen concordancia con las métricas existentes y no se hace una retroalimentación con

toda la información. Bajo estas condiciones se realizan las pruebas en el departamento, lo que trae como consecuencia que su realización sea mayormente intuitiva, donde el especialista revisa el software a través de un conjunto de pruebas que pone en práctica con el objetivo de emitir un criterio de si es liberado o no el producto, esto está sujeto al grado de experticia de este especialista, por lo que el proceso que se realiza no tiene el nivel necesario.

Por toda esta situación problemática, se desea mejorar esto a través de la implantación de un nuevo proceso de evaluación en el Departamento. El que permitirá después de realizarse las pruebas poder hacer uso de la información que en ese proceso se genera de la manera más detallada, organizada y exigente. Esto traerá como consecuencia gran cantidad de plantillas, para lograr obtener medidas cualitativas y cuantitativas de la calidad del producto con las cuales se utiliza la aplicación de una amplia gama de métricas, con un conjunto de fórmulas asociadas, que se deberán aplicar a varios proyectos al mismo tiempo por lo que sería demasiado engorroso y lento aplicarlo de forma manual, pero éste por si solo no es capaz de imponerse, por lo que es necesario poseer un medio o herramienta que obligue a todos ajustarse a él, elevando la eficiencia del mismo. Debido a esto se desea implantar una herramienta automatizada que apoyará el proceso de evaluación.

Por lo antes descrito determinamos como nuestro **problema científico**: ¿Cómo brindar apoyo al nuevo proceso del Departamento de Pruebas?

Definiéndose como **Objeto de Estudio**: Los procesos del Departamento de Prueba.

Enmarcándose como **Campo de Acción**: El proceso de evaluación de software del Departamento de Prueba de la UCI.

**Objetivo General**: Desarrollar una herramienta que sirva de apoyo al nuevo sistema de gestión del proceso para la evaluación de la calidad de un software Departamento de Prueba.

Entre los **posibles resultados** se encuentra el modelado e implementación de una herramienta que apoye el proceso de evaluación de software del Departamento de Prueba.

La **Idea a defender** que se plantea es que, con el desarrollo de una herramienta automatizada, se apoyará el proceso de evaluación del software que se lleva a cabo en el Departamento de Prueba y se facilitará el proceso de evaluación de la calidad de los productos que se desarrollan en la Universidad.

### **Tareas a desarrollar para cumplir los objetivos**

- Investigación del proceso de evaluación en el Departamento de Prueba.
- Investigación y estudio de los estándares para el diseño de aplicaciones Web.
- Estudio y selección de la metodología, herramientas y lenguaje del modelado que se utilizarán para el desarrollo de la aplicación.
- Levantamiento de las necesidades para la implementación de la aplicación para automatizar el proceso de Evaluación de Software.
- Análisis, diseño y modelación de la aplicación.
- Implementación de una Herramienta para la Gestión del Proceso de Evaluación de Software de los Centros de Producción

### **Métodos Científicos de Investigación:**

Se llevaron a cabo diferentes métodos científicos para realizar la investigación. Como método Teórico se utilizó el **Analítico-Sintético** pues uno de los primeros pasos para llevar a cabo la presente tesis es buscar y analizar documentos e información sobre los estándares internacionales para la evaluación de Software y extraer los elementos más importantes relacionados con el objeto de estudio. Otro método teórico que se aborda en la investigación es el método **Histórico-Lógico** para analizar a nivel internacional y nacional el empleo de sistemas informáticos similares al que se propone, así como investigaciones realizadas anteriormente sobre el tema.

Dentro de los métodos Empíricos se usó la **Entrevista**, para de esta forma recoger toda la información referente a las necesidades actuales y perspectivas que debe cumplir el portal.

El trabajo está estructurado de la siguiente manera:

**Capítulo 1:** Fundamentación teórica: En este capítulo se realiza un estudio del estado del arte revisando algunos de los estudios y sistemas más utilizados para la evaluación de productos exponiendo las ventajas y las desventajas de cada uno. Se realizará un análisis de las metodologías, herramientas y lenguajes que existen en la actualidad y que pudieran ser útiles en el desarrollo de la solución justificando la selección de los mismos.

**Capítulo 2:** Características del sistema: Se presentará una descripción general de la propuesta de sistema, se determinarán los conceptos fundamentales para realizar el modelo de dominio, se identificarán los requisitos funcionales y no funcionales del sistema a desarrollar y se describirán en términos de casos de usos.

**Capítulo 3:** Análisis y diseño: Se realizará el análisis y diseño donde se obtendrán los artefactos necesarios para la posterior implementación del componente entre los que podemos mencionar los diagramas de clases del análisis y del diseño y los diagramas de interacción. En este capítulo además se determinará la arquitectura y los patrones de diseño a utilizar en la implementación.

**Capítulo 4:** Implementación: Se llevará a cabo la implementación de la aplicación, obteniéndose los artefactos correspondientes como son el diagrama de despliegue.

Por último se tienen las conclusiones, las recomendaciones, referencias bibliográficas, bibliografías y anexos.

## **CAPÍTULO 1: FUNDAMENTACIÓN DEL TEMA**

---

### **INTRODUCCIÓN**

---

Actualmente merece mucha importancia la interfaz gráfica de usuario a la hora de crear aplicaciones informáticas y para desarrollar las mismas se utilizan los framework de desarrollo que más ventajas proporcionan. Se tienen en cuenta también, las tendencias actuales para el desarrollo de aplicaciones con el objetivo de lograr un producto lo más adecuado posible a los requerimientos de los clientes, en el menor tiempo, además de estar basado en las tecnologías y métodos de desarrollo más actualizados y seguros.

### **APLICACIONES WEB**

---

Antes de entrar en los detalles propios de la aplicación a desarrollar, cabe destacar, que para cumplir los objetivos de capturarlos durante el levantamiento de necesidades, se debe desarrollar un sistema que les posibilite a múltiples usuarios tener acceso al mismo tiempo a la aplicación, posibilitando la obtención rápida de la información, sin necesidad de compartir escritorios, ni del intercambio físico directo entre los administradores y sus usuarios, lo que no es posible mediante aplicaciones de escritorio. En este caso, la opción óptima es el uso de una aplicación Web.

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet. Estas son populares debido a la practicidad del navegador Web como cliente ligero. Ejemplo de estas son los web mails, wikis, web log y tiendas en línea. (Corsi, 2010)

---

### **VENTAJAS DEL USO DE APLICACIONES WEB**

---

Además de la ventaja de posibilitar el acceso a varios usuarios simultáneamente al sistema, estas poseen un gran número de características, entre las que se encuentran las que se muestran a continuación:

- Las aplicaciones Web, son actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación.

# Capítulo 1: Fundamentación del Tema

- Tienen un precio menor que las aplicaciones de escritorio, estas no requieren la infraestructura de distribución, soporte técnico y marketing requerido por las aplicaciones de software tradicionales. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes.
- Los datos están online, es decir, no es necesario encontrarse precisamente en la computadora en la que se instaló el software para tener acceso al mismo, sin necesidad de compartir escritorios. (Corsi, 2010)

---

## TIPOS DE APLICACIONES WEB

---

Las aplicaciones Web se dividen en tres grupos fundamentales: Informativas, De negocio y Portales.

- **Las aplicaciones Web informativas**, son las más sencillas de realizar. En las que sólo se muestra información de un tema determinado, donde la página principal es la de mayor importancia a la hora de su elaboración, ya que es la que da la idea del contenido que se encontrará en la aplicación.
- **Las aplicaciones Web de negocio**, incluyen las aplicaciones informativas y opciones donde el cliente selecciona lo que necesita de un producto o servicio, en esta el intercambio de información es más complejo, en su forma más sencilla, luego de la selección de los elementos que el usuario necesita, el sistema le pide los datos y los almacena en la base de datos con toda la información ingresada por el usuario. (Corsi, 2010)
- **Los portales** son el grupo más complejo de las aplicaciones Web. Un portal de Internet es conocido como “Puerta o puerto”, es decir, es un sitio Web que sirve como punto de partida para la navegación en la red. Posibilitando de forma fácil sencilla, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse facilidades para la localización de sitios como son los buscadores, noticias, buzones de correo, directorios telefónicos y foros de discusión con temas de interés general, además principalmente están dirigidos a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios de a una institución pública o privada.

---

## TIPOS DE PORTALES

---

# Capítulo 1: Fundamentación del Tema

El principal objetivo de los portales es asociar al usuario y que utilicen el portal de forma continuada o diaria. De esa forma el portal gana tráfico constante y alto para poderse autofinanciar con publicidad.

Los portales tienen tres pilares fundamentales para atraer al usuario y son:

- Información (Buscadores, directorios, noticias, catálogos y servicios)
- Participación (E-mail, foros, Chat)
- Comodidad (En un solo espacio lo tienes todo a mano)

El mundo de Internet ha llegado a tal punto que los portales se han clasificado por tipos (horizontales, verticales, específicos, corporativos y móviles).

## **Portales Horizontales**

Es un portal de carácter general orientado a todo tipo de usuario, ofreciendo contenidos que son demandados por el público. Incluye servicios como comunidades virtuales, Chat, e-mail, espacio Web gratuito, grupos de discusión, noticias e información. Son portales para usuarios no muy experimentados en el mundo de Internet, de esa forma le das todas sus necesidades en un solo sitio teniéndolos motivados.

## **Portales Especializados**

Cada vez son más los usuarios que demandan sitios en los que se pueda encontrar la mayor cantidad de información sobre un tema en específico, los portales horizontales y verticales no les satisfacen por tener contenidos demasiados generales o superficiales para ellos. Por ese motivo se han creado portales exclusivos en temas específicos como, educación, zonas geográficas, salud, etc.

## **Portales Verticales**

Los portales verticales son sitios Web agrupados por áreas de interés o servicios específicos. Representan un acceso directo a Internet, capaz de ofrecer argumentos seleccionados, especializados y exhaustivos y será aquí donde las empresas encuentren nuevas oportunidades de promoción. Portales verticales, se dirigen a usuarios para ofrecer información de un determinado tema como puede ser un portal de música, empleo, inmobiliario, un portal de finanzas personales, arte, deportes, etc. (MANJARRES V, (04/2009))



# Capítulo 1: Fundamentación del Tema

---

## **ANÁLISIS DE OTROS PORTALES EXISTENTES**

---

En la actualidad va en incremento las empresas que incursionan en el negocio de prestar servicios mediante aplicaciones Web. Algunas de ellas centran su atención en construir herramientas que permitan automatizar procesos que hasta el momento se muestran de difícil y lento desempeño. Basadas la gran mayoría en los sistemas certificados de la industria de software a partir de las normas de calidad, estas empresas se proponen diseñar metodologías totalmente documentadas, flexibles y completas, que establecen el ciclo de vida de los procesos y que constituyan una herramienta contrastada para asegurar la calidad de los resultados esperados.

Sin embargo, cada vez se tiene más presente la necesidad de garantizar un determinado nivel de calidad del software informático. Los clientes exigen el aseguramiento de la calidad mediante la certificación y por otro extremo la existencia de una serie de atributos que pueden ser medidos de forma automática, ayudan a su cumplimiento. Bajo este aspecto en el mundo existen algunos sitios que ponen su empeño en desarrollar sistemas automatizados que evalúen el cumplimiento de determinado criterio que resulte prioritario en un sistema.

### **A NIVEL INTERNACIONAL**

---

**Sitio del Laboratorio de Tecnologías de la Información del NIST:** El software comienza a ser una preocupación de las agencias reguladoras de equipos médicos, pues comienzan a reportarse incidentes atribuibles al software médico, por lo que se hace necesario poder comprobar el correcto funcionamiento de los software instalados en estos equipos, es por esto que se crea un esquema de certificación basado en la norma ISO 9000, y la guía ISO 9003. En este último caso los criterios de seguridad resultan prioritarios. (U.S. Commerce Department, 2003)

# Capítulo 1: Fundamentación del Tema

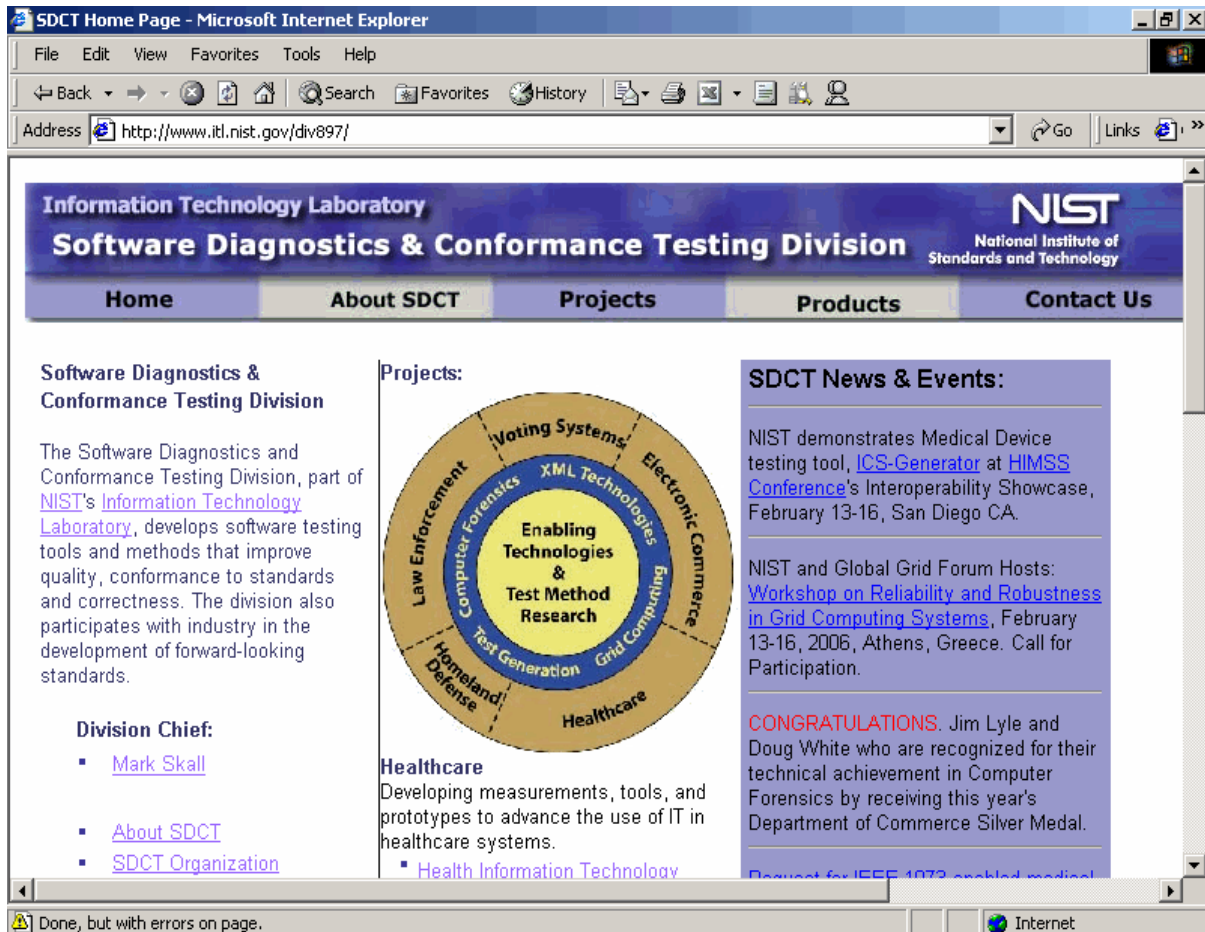


Figura 1. Sitio del Laboratorio de Tecnologías de la Información del NIST.

**Anaconda:** (analizador automático de código Natural para el desarrollo de aplicaciones) es un sistema para la evaluación de la calidad del software de aplicaciones informáticas desarrolladas en el entorno Natural/ADABAS. Analiza la aplicación partiendo del código fuente, verificando que cumple los requisitos de calidad establecidos en la instalación para poder ser implantada y explotada, y proporciona un diagnóstico que refleja el nivel de calidad del software. Este diagnóstico lo realiza basándose en unas métricas y pesos definidos a partir del conocimiento de los expertos. Es capaz de analizar miles de líneas de código fuente por minuto generando un informe detallado con los errores y recomendaciones. (Sistema ANACONDA para el análisis automático de la calidad del software, 2009)

# Capítulo 1: Fundamentación del Tema

## **A NIVEL NACIONAL**

---

Se realizaron búsquedas al respecto de aplicaciones y estudios a nivel nacional y no se han encontrado resultados acerca del tema, se espera que este estudio marque pautas en el tema.

## **ASPECTOS IMPORTANTES PARA CREAR UNA WEB PARA EVALUAR LA CALIDAD DE UN PRODUCTO SOFTWARE**

---

Lo principal para la realización de un sistema, es estar convencido que no se puede conseguir un liderazgo tecnológico sólido si no va acompañado de la excelencia de los resultados. Es importante tener en cuenta que tanto la constante evolución de las tecnologías como las continuas publicaciones de las familias de normas; atentan contra la perdurabilidad y eficiencia del sitio, por lo que se debe tener en cuenta en el momento de su realización para que pueda ser fácilmente actualizado sin provocar cambios en el diseño de la Web, por ultimo es necesario diseñar una estructura Web en base a una navegación simple e intuitiva para el usuario, facilitándole la labor que pretendemos que realice.

## **TECNOLOGÍAS Y HERRAMIENTAS PROPUESTAS PARA EL DESARROLLO DEL SISTEMA.**

---

Una preocupación cada vez mayor de los ingenieros de software es la necesidad de mantenerse al corriente de la rápida evolución de las tecnologías. Resulta de gran importancia mantenerse actualizado en cuanto a los framework de desarrollo de aplicaciones, estándares de comunicación, así como de las principales tendencias en el uso de estas tecnologías, para poder desarrollar una aplicación con calidad, en el tiempo requerido y que satisfaga las necesidades de los clientes.

## **METODOLOGÍA DE DESARROLLO**

---

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Indican, paso a paso, todas las actividades a realizar para lograr el producto informático deseado, además de las personas que deben participar en el desarrollo de las actividades y qué papel deben tener.

# Capítulo 1: Fundamentación del Tema

Además se encargan de elaborar estrategias de desarrollo de software que promueven prácticas centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente. (Wesley, 2005)

---

## **METODOLOGÍA ÁGIL XP (EXTREME PROGRAMMING).**

---

La metodología XP permite establecer iteraciones muy cortas, apropiada para un entorno caracterizado por requerimientos cambiantes. Su objetivo principal es tener una nueva versión a cada instante, mostrarlo al cliente, ver lo que opina y seguir programando, tener una comunicación fluida con el cliente y el usuario final, es decir, define una manera de reunir a clientes y programadores en un equipo firmemente integrado con condiciones de trabajo que promueven la comunicación y solución de un problema. Se ha clasificado como una metodología ágil, ya que plantea aumentar constantemente la velocidad del proyecto.

Esta metodología reduce la documentación asociada a cada proyecto (esto puede ser riesgoso con proyectos de complejidad alta). No cuenta prácticamente con una documentación para su constancia, lo que implica que los desarrolladores se acostumbren a esta mala práctica. Estas son las principales razones por las que no se seleccionó esta metodología para el desarrollo de la aplicación. (Descripción y características de este potente y moderno, 2001)

---

## **METODOLOGÍA DE DESARROLLO RATIONAL UNIFIED PROCESS (RUP)**

---

RUP es una metodología de desarrollo de software orientada a objetos que proporciona un método disciplinado para asignar las tareas y responsabilidades dentro del equipo de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que resuelva las necesidades del usuario dentro de un cronograma predecible y al menor costo posible. Está basado en componentes y utiliza el lenguaje UML para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Esta metodología es ideal para la gestión de los requisitos, desarrollo visual del software (con UML), la verificación continua de la calidad del software, la gestión de los cambios y sobre todo, para proyectos grandes, a largo plazo y con un equipo de desarrollo numeroso. (Jacobson, I, Booch, G., 1999)

---

## **JUSTIFICACIÓN DE LA METODOLOGÍA SELECCIONADA.**

---

# Capítulo 1: Fundamentación del Tema

Por disímiles razones se escoge la metodología RUP para la realización de este trabajo, por su nivel al describir los procesos y crear diagramas UML. Es muy conveniente porque durante su ciclo de vida se desarrolla toda una metodología iterativa e incremental que va eliminando los errores cometidos en las iteraciones previas, logrando que al final del proceso se obtenga como resultado un producto de calidad. Aunque este trabajo cuenta de pocos procesos, algunos de estos son complejos y rigurosos por lo que se hace necesario el uso de una metodología potente que administre el tiempo y costo del proyecto y permita controlar la calidad del mismo a través de sus iteraciones.

## **HERRAMIENTAS PARA LA MODELACIÓN VISUAL**

---

Las herramientas CASE (Computer Aided Software Engineering), utilizan Unified Modeling Language (UML). Pueden ser generalmente aplicadas a cualquier sistema o colección de herramientas que permita automatizar el proceso de diseño y desarrollo de software. Estas herramientas están destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste en términos de tiempo y de dinero. 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, implementación de parte del código automáticamente con el diseño dado, compilación automática y documentación o detección de errores entre otras. (Jacobson, I, Booch, G., 1999)

---

## **VISUAL PARADIGM**

---

Visual Paradigm es una herramienta libre utilizada para el modelado de aplicaciones, como lenguaje de modelado utiliza UML y está diseñada para una gran cantidad de usuarios. Esta herramienta permite construir la aplicación con mayor rapidez, mayor trabajo en equipo, fácil de utilizar, mayor exactitud, además de facilitar la interoperabilidad con otras herramientas CASE, la mayoría de los IDEs<sup>1</sup> principalmente y permitir la integración de todos los componentes.

---

<sup>1</sup> IDE (Integrated Development Environment - Entorno integrado de desarrollo). Aplicación compuesta por un conjunto de herramientas útiles para un programador.

# Capítulo 1: Fundamentación del Tema

---

## **RATIONAL ROSE**

---

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML, que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo de un sistema informático. Brinda facilidades para la generación de la documentación del software que se está desarrollando y posee un gran número de estereotipos predefinidos que agilizan el proceso de modelación. Rational Rose utiliza un proceso de desarrollo iterativo controlado donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración. Rose permite que hayan varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. (Jacobson, I, Booch, G., 1999)

---

## **JUSTIFICACIÓN DE LA HERRAMIENTA PARA LA MODELACIÓN VISUAL**

---

Después del estudio hecho se decidió utilizar Visual Paradigm que ayuda a la comunicación entre los miembros de equipo. Emplea una respuesta rápida y bajos requisitos de memoria del motor de persistencia, lo que le permite manejar grandes y complicadas estructuras de un proyecto en una forma altamente eficiente y sólo requiere de una configuración de escritorio.

---

## **LENGUAJE DE REPRESENTACIÓN VISUAL PARA EL MODELADO DEL SISTEMA.**

---

El UML (Unified Modeling Language) está basado en el uso extensivo del método Booch, OMT y Jacobson; es decir, el UML es la evolución de éstas y otras aproximaciones para modelado de procesos de negocios, objetos, y el modelado de componentes.

El Lenguaje Unificado de Modelado establece un conjunto de notaciones y diagramas estándar para la modelación de los sistemas orientados a objetos, y describe la semántica de lo que significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y

# Capítulo 1: Fundamentación del Tema

organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas entre los más importantes:

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Implementación para modelar la distribución del sistema. (Diseño del Sistema de tarjetas de Crédito con UML, 2003)

Debido a todos estos aspectos se decidió utilizar UML como lenguaje de representación visual, ya que es además el utilizado por RUP para el desarrollo visual del software.

## **LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS PARA LA WEB.**

---

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial. Para desarrollar una aplicación como la que nos proponemos, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (Uniform Resource Locators), no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de Internet, como es el HTTP (Hypertext Transfer Protocol).

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML (Hypertext Markup Language). (Luke Welling, Laura Thomson, 2009)

---

## **PHP**

---

# Capítulo 1: Fundamentación del Tema

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Ventajas:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos o extensiones. (Ped,2008)

---

## **JAVA**

---

Es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la Compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Ventajas:

- Es un lenguaje relativamente sencillo y fácil de aprender.
- Es muy productivo, con poco código se puede hacer mucho más que con la misma cantidad de código de C ó C++
- Es un Lenguaje de Programación Orientada a Objetos lo que permite organizar el código en entidades como las clases compuestas de datos y funciones, y a través de la característica de la herencia se puede organizar las clases en jerarquías. (Descripción y características de este potente y moderno, 2001)



# Capítulo 1: Fundamentación del Tema

---

## GROOVY AND GRAILS

---

Groovy es un lenguaje dinámico para la JVM<sup>2</sup>, sus creadores lo conciben como: "...una súper versión de Java. Puede explotar las capacidades empresariales Java pero además tiene características frescas de productividad como los closures...", Groovy está basado en paradigmas como 'Convención sobre configuración', soporta DSL (Domain Specific Languages), simplifica las pruebas, además las docenas de líneas de código en Java pueden acortarse a pocas líneas de código en Groovy ganando en legibilidad, mantenibilidad y eficiencia.

"Grails apunta a brindar el paradigma 'Convención sobre configuración' de Groovy. Es un framework Open Source para aplicaciones Web que potencializa el lenguaje Groovy y complementa el desarrollo Java en Web. Se puede usar Grails como un entorno de desarrollo independiente que esconde todos los detalles de la configuración o integra la lógica del negocio hecha en Java. Grails apunta a hacer el desarrollo tan simple como sea posible y con esto sería muy atractivo para un amplio rango de desarrolladores no solo para aquellos que estén en la comunidad Java además no centra su desarrollo en reinventar lo bueno, sino mejorarlo: trata de tomar lo mejor de las tecnologías y adaptarlas mucho más eficientes. Por ejemplo, Hibernate es un buen ORM (object- relational mapping), con poderosas y avanzadas herramientas. Grails crea un simple DSL, que simplifica el trabajo con Hibernate, dejando fuera las complicadas configuraciones de mapeo en archivos XML. (Beginning, et al., 2009)

---

## JUSTIFICACIÓN DE LA LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS PARA LA WEB

---

Después del estudio hecho se decidió utilizar el lenguaje Groovy and Grails, ya que este utiliza una sintaxis muy parecida a Java y comparte el mismo modelo de objetos. Además que la programación asegura la seguridad de sus clases y objetos. Desde Groovy se puede acceder directamente a todas las API<sup>3</sup> existentes en Java. El byte code generado en el proceso de compilación es totalmente compatible

---

<sup>2</sup> Java Virtual Machine (Maquina Virtual de Java)

<sup>3</sup> 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. Usados generalmente en las bibliotecas.

# Capítulo 1: Fundamentación del Tema

con el generado por el lenguaje Java para la JVM, por tanto puede usarse directamente en cualquier aplicación Java. Todo lo anterior unido a que la mayor parte de código escrito en Java es totalmente válido en Groovy hace que este lenguaje sea de muy fácil adopción.

Grails como herramienta para la Web ha supuesto un gran paso adelante para el framework, al corregir muchos de los pequeños detalles que aún quedan por pulir, fundamentalmente en la experiencia de desarrollo.

Además una de las potencialidades de Grails es su bajo costo de instalación ya que solo necesita que estén instalados dos elementos: la maquina virtual de java (JVM) y el framework Grails 1.1 o versiones superiores. Mientras que para poder desarrollar un proyecto web en Java habría que instalar una serie de elementos como: Java, algún servidor de bases de datos, Spring, Hibernate, algún framework de desarrollo web.

## **SISTEMA DE BASE DE DATOS**

---

Los sistemas de bases de datos están diseñados para gestionar grandes volúmenes de información. Un sistema de bases de datos tiene como objetivo simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean muy reducidos.

Para lograr un completo ambiente de desarrollo Groovy, requiere de una base de datos relacional, para ello incluye una base de datos 100% Java llamada HSQLDB, la cual se puede usar como un servidor de bases de datos autónomo o como una base de datos incluida y se puede configurar para que esta corra en la memoria de la PC o persista en el disco duro. También da la posibilidad de configurarse para que use otra base de datos como MySQL. (Graeme, et al., 2009)

## **PROPUESTA DE SOLUCIÓN**

---

Después de haber realizado un estudio de las tendencias actuales y de las principales características de las tecnologías más usadas en la actualidad, se propone la creación de una herramienta que brinde soporte al nuevo Sistema de Gestión para la Evaluación del Software en el Departamento de Pruebas. Se

# Capítulo 1: Fundamentación del Tema

considera que se adoptara un Portal Vertical por la facilidad que brindan pues son sitios web agrupados por áreas de interés o servicios específicos. Utilizando Groovy como lenguaje de programación y Grails como framework para desarrollo de aplicaciones web, debido a los grandes beneficios que aporta, además de la rapidez y flexibilidad en su programación, además de contar con una amplia comunidad de programadores, a nivel mundial, lo que permite acceder relativamente fácil a su documentación por parte de sus usuarios.

Como sistema gestor de base de datos: HSQLDB, ya que es el gestor que viene acoplado a la tecnología Groovy and Grails. Como metodología de desarrollo se empleara RUP, como herramienta para la Modelación Visual utilizaremos Visual Paradigm y como Lenguaje de Representación Visual para el modelado del sistema, UML. Otra de las razones para la selección de las herramientas a utilizar, es que además de todas las ventajas mencionadas anteriormente, estas son las herramientas que se ajustan a la arquitectura para los sistemas que conforman la intranet universitaria, que plantea la Dirección de Informatización de la Universidad de las Ciencias Informáticas (UCI), perteneciente a la Infraestructura Productiva (IP) en la UCI.

# Capítulo 2: Características del Sistema

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

---

### **INTRODUCCIÓN**

---

En este capítulo se abordará todo lo referente a las características del sistema y el modelado de los procesos para su mejor comprensión. Además se detallan los requisitos tanto funcionales como no funcionales a cumplir en la realización de la misma.

### **PROBLEMA**

---

Actualmente en el Departamento de Prueba, es necesario contar con un portal que sirva de herramienta para brindar soporte al nuevo proceso de Gestión de la Evaluación del Software y brinde la mayor cantidad de información referente a dicha área. Esto permite que el número de personas implicadas en el empleo de esta herramienta sea cada vez mayor. En la actualidad el Departamento, desarrolla su labor con una serie de ineficiencias, la principal y más grave es que la evaluación se hace de forma cualitativa lo que reduce la posibilidad de establecer un buen nivel a los productos que se desarrollan en la Universidad. Por lo que se hace indudable la falta de una aplicación que permita apoyar el nuevo proceso que se desea implantar y permita visualizar la evaluación cuantitativa del producto en cuestión, además de aportar una fuente de apoyo al resto de las empresas que deseen implantar sistemas de evaluación en sus negocios.

### **OBJETO DE AUTOMATIZACIÓN**

---

La herramienta propuesta, permitirá automatizar el proceso de Evaluación de Software que se implantará en el Departamento de Pruebas. Todo producto para su liberación será evaluado cuantitativamente por la herramienta. El personal implicado podrá tener información sobre el proceso de evaluación, además podrá visitar el sitio y encontrar en la página principal del mismo, un menú central que le permitirá acceder a los puntos más generales del entorno.

# Capítulo 2: Características del Sistema

## **INFORMACIÓN QUE SE MANEJA**

---

La información que se maneja es precisa y exacta, como son: artefactos, características, sub características, modelos de pesos, plan de evaluación, modelos de calidad, expedientes de productos-proyectos, altas de liberación entre otros.

## **MODELO DE DOMINIO**

---

Una de las primeras actividades centrales de un ciclo de desarrollo consiste en crear un modelo conceptual para los casos de uso, en el cual se explican a sus creadores los conceptos significativos en un dominio del problema. (Schmuller, 2000)

El Modelo de Dominio también conocido como Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Este representa clases conceptuales del dominio del problema, conceptos del mundo real y no de los componentes de software. Una clase conceptual también conocida como entidad puede ser una idea o un objeto físico (símbolo, definición y extensión).

El modelo de dominio se representa en UML con un diagrama de clases en el que se muestra: conceptos u objetos del dominio del problema: clases conceptuales y las relaciones entre las mismas. (RÍOS,1999)

---

## **JUSTIFICACIÓN DE LA UTILIZACIÓN DEL MODELO DE DOMINIO**

---

Al ejecutar la entrevista con el personal involucrado en el proceso de prueba del Departamento, comprobamos que se hacía difícil modelar el negocio ya que las actividades y roles que intervienen en el proceso no contaban con una descripción clara, ni con una estrategia bien definida, ya que muchos aspectos se dejaban a la intuición de los involucrados con las pruebas, por lo que nos planteamos la tarea de describir el funcionamiento de la aplicación mediante una serie de conceptos y descripciones, materializándolos en la definición de entidades relacionadas entre ellas y agrupándolos en un modelo de dominio con el fin del facilitar el entendimiento del proceso actual.

---

## **DESCRIPCIÓN DEL MODELO DE DOMINIO**

---

## Capítulo 2: Características del Sistema

En la actualidad el centro funciona de la siguiente forma: el Proyecto hace la solicitud de revisión, la cual es valorada por la Dirección de Calidad de la Universidad, luego esta unidad convoca a una reunión de inicio con el personal del proyecto involucrado con la solicitud, en esta reunión se pacta todo lo referente a la revisión, se realizan pruebas exploratorias, si el producto no cumple con los Criterios de Criticidad<sup>4</sup> se declara prueba fallida y se anuncia al proyecto que no se puede continuar con la prueba, sino se le realizan al entregable pruebas masivas que conllevan a listar las no conformidades, culminando con la liberación cuando la no conformidades son nulas o mínimas; pactadas con el cliente de antemano.

---

### REGLAS

---

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, puesto que regulan algún aspecto del negocio. El proceso de especificación implica que hay que “identificarlas” dentro del negocio, “evaluar” si son relevantes dentro del campo de acción que se está modelando e “implementarlas” en la propuesta de solución”. (Pressman, 1999)

Durante la entrevista se lograron identificar dos eventos que muestran importancia en la realización del proceso actual:

- Se declara prueba fallida al entregable que no cumpla con los criterios de criticidad.
- Se debe de realizar una reunión de inicio antes de comenzar con la evaluación.

---

### ACTORES Y TRABAJADORES

---

“Un actor es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados”. (Pressman, 1999)

A continuación se muestran los actores en la siguiente tabla.

Actores	Justificación
Equipo de Desarrollo.	Personal encargado del desarrollo de la aplicación.

---

<sup>4</sup> Establecen los parámetros para declarar un producto del desarrollo software en estado crítico de terminación

## Capítulo 2: Características del Sistema

Directivos	Personal interesado en el estado del proyecto y su avance, así como en el resultado de las evaluaciones.
------------	--

**Tabla 1.** Actores del Dominio.

“Un trabajador representa un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades.” (Pressman, 1999)

A continuación se muestran los trabajadores en la siguiente tabla.

Trabajadores	Justificación
Especialista	Diseña, elabora y ejecuta las evaluaciones.
Probadores	Ejecutan las evaluaciones y elaboran el Registro de No Conformidades <sup>5</sup> .

**Tabla 2.** Trabajadores del Dominio.

---

### DIAGRAMA DE CLASES DEL MODELO DE DOMINIO

---

---

<sup>5</sup> Documento donde quedan registrados el conjunto de defectos que presenta un entregable así como la evaluación otorgada en escala a 5

## Capítulo 2: Características del Sistema

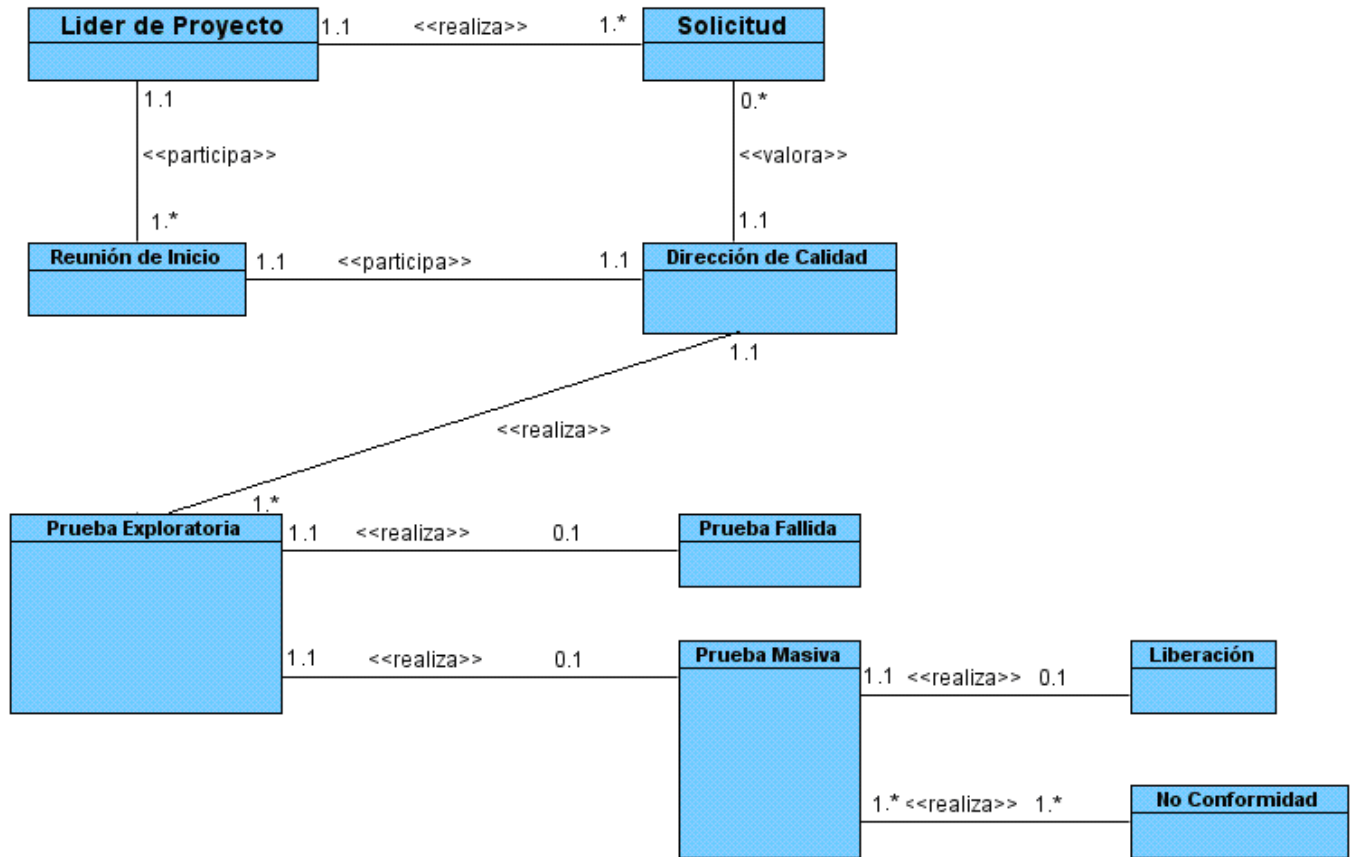


Figura 2. Diagrama de clases del Dominio.

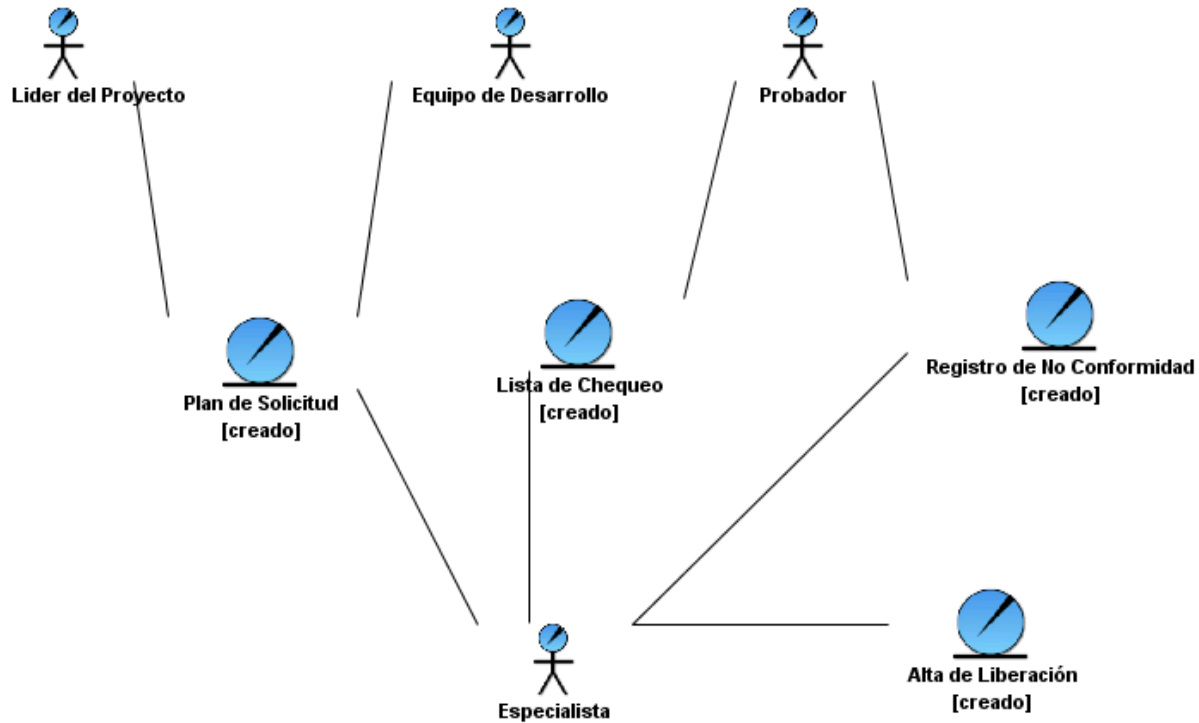
---

### MODELO DE OBJETOS

---



## Capítulo 2: Características del Sistema



**Figura 3.** Modelo de Objeto

---

### ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE

---

En este flujo se establece lo que tiene que hacer exactamente el sistema que se va a construir. En esta línea los requisitos son el contrato que se debe cumplir, de manera tal que los usuarios finales comprendan y acepten los requisitos que se especifiquen. Se dividen en dos grupos: los requisitos funcionales y los requisitos no funcionales. (Pressman, 1999)

---

#### REQUISITOS FUNCIONALES

---

Los requisitos funcionales son aquellas condiciones o capacidades que el sistema debe cumplir. Los requisitos funcionales deben comprenderlo tanto los desarrolladores como los usuarios. En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización. Estos son el punto de partida para identificar qué debe hacer el sistema. (Pressman, 1999)

## Capítulo 2: Características del Sistema

A continuación se detalla la propuesta del proceso de Evaluación para el Laboratorio de Pruebas:

Primeramente se **Entrega la Solicitud de Evaluación**: El proceso de evaluación se iniciará con la entrega de una Solicitud de Evaluación a la entidad evaluadora, artefacto donde se formaliza las intenciones de un proveedor de que su producto sea evaluado.

Aceptada la solicitud, se debe convocar a una **Reunión Negociación**, o encuentro formal, donde se le presente a dicho proveedor, las características del proceso completo, sus peculiaridades, las responsabilidades a cumplir por cada parte, así como se determinarán los intereses de evaluación.

En este instante se procede a **RF1 Crear el Expediente de Evaluación**: Es de vital importancia que por cada producto-proyecto que se presente a revisar se cree un Expediente, que recogerá toda la información, dígase nombre del proyecto al que pertenece, nombre del producto, ya que un proyecto puede presentar varios productos, así como datos de la persona representante del entregable. Es importante que esta funcionalidad pueda ser consultada, modificada o eliminada de ser necesario y permitido.

Cuando evaluamos la calidad de un software es necesario establecer la evaluación de los requerimientos, especificar la evaluación, designar esta y finalmente ejecutar la evaluación.

Por esta razón es importante **RF5 Establecer Requisitos**: Uno de los objetivos fundamentales de una evaluación radica en una futura comparación con otros productos, donde se pueda tener una orientación hacia qué tan mejor o peor es un producto con respecto a otro y así tomar decisiones. Dada la diversidad de soluciones informáticas y sus características específicas, resulta desacertado hacer comparaciones sin tener en cuenta determinados elementos. (Por ejemplo: evaluar la seguridad informática entre una multimedia educativa para niños y el sistema de monitoreo de una torre de control de un aeropuerto, evaluar la usabilidad entre una tienda Web y sistema de escritorio para la gestión de trazas de señales radiales, evaluar la portabilidad entre un sistema de enseñanza multimedia y una aplicación de gestión bancaria, etc.)

Por ello, es importante que la parte evaluadora posea una clasificación previa, donde agrupe los productos a evaluar por un criterio determinado, y que esta clasificación se mantenga como hilo rector. También es

## Capítulo 2: Características del Sistema

importante **RF5.1 Especificar el modelo de calidad**: Existen varios modelos de calidad, desde el de McCall y el de Bohem, hasta los más actuales como la ISO/IEC 9126 o la construcción (año actual 2010) de la ISO/IEC 25010. En todos ellos se presenta una estructura de árbol donde se presentan las características que los definen y componen. Se debe seleccionar, previamente por la parte evaluadora, cuál de estos modelos va a ser el modelo rector de sus evaluaciones. El más recomendado actualmente (2010) es la ISO/IEC 9126. (Joaquín, 2010)

La parte evaluadora comenzará trabajando con un modelo en específico; en el futuro pudiera ir agregando otros más, aunque no es lo más recomendable a menos que sea para sustituir paulatinamente por uno más actualizado.

Otro aspecto importante es **RF6 Confeccionar Plan de Evaluación**: Concluidas las actividades de definición, es necesario concretar todo ello en una serie de acciones específicas que tributen a lo ya establecido. Introducir la información referente a la métrica a usar (nombre, descripción o propósito, fórmula), los datos a recolectar, que no es más que introducir las variables que componen la fórmula de la métrica, explicando su significado. El objetivo fundamental del Plan de Evaluación es determinar la estrategia a seguir para recolectar dichos datos. Es importante determinar un cronograma de acciones específicas y los recursos necesarios ya con todos los elementos fundamentales definidos, se elaborará un cronograma, donde se definan las fechas y actividades a realizar, con sus responsables.

Ya definidos los Requisitos y el Plan de Evaluación debemos **RF7 Llenar el Modelo de Calidad**: Esta funcionalidad consiste en darle valor establecido del 1 al 10 a cada sub-característica, permitiendo establecer cuantitativamente la incidencia de esa cualidad en el producto evaluado.

Luego es importante **RF8 Llenar el Modelo de pesos para la evaluación**: En orden de evaluación de software es necesario el uso de un modelo de peso que rompa con la idea de la calidad del software bajo características diferentes. Pero si tener claro que, dentro de una evaluación, unas características van a poseer más importancia que otras, estos estaría determinado por el grupo al que pertenezca dentro de las clasificaciones declaradas, es decir, que por cada clasificación debe existir un modelo de pesos específico. El modelo de peso estaría compuesto por un grupo de coeficientes que tendría cada característica y sub característica, evidenciando cuantitativamente esta diferencia de importancia.

## Capítulo 2: Características del Sistema

También debe ser posible **RF9 Establecer nivel de aceptación de la conformidad** Como elemento previo, se debe establecer cuál será el nivel de aceptación de la evaluación. Para ello se deben cumplir los siguientes pasos: **9.1 Determinar el valor máximo a obtener en una evaluación** (para el caso de no poder, revisar el sistema de modelo de pesos y los rangos de numéricos de las métricas, para ser ajustados en función de esto) y **9.2 Dividir ese rango tres partes básicas:** inaceptable, mínima aceptable y aceptable.

Otro punto es **RF10 Ejecutar la Evaluación:** Iniciado la realización del cronograma de actividades de evaluación se debe ser muy riguroso con ello. Debe existir alguien que dirija, controle y monitoree el proceso, de manera tal que en caso de eventualidades, se tomen las medidas pertinentes. En esta actividad lo fundamental es ejecutar todo lo planificado y recolectar los datos de las mediciones realizadas. Es de vital importancia que esta funcionalidad se pueda ejecutar en el mismo instante en que se cree el Expediente pues esta es la esencia de la realización del mismo

**RF11 Realizar Ranking por Solicitud** es otra de las funcionalidades que se desea implementar, para de solicitarse por la dirección del proyecto al que pertenece determinado producto, este sea ubicado en un Ranking con otros productos ya evaluados y así se podrá tener una visión del nivel de competencia de acuerdo a un parámetros determinado.

Se desea **RF12 Realizar Comprobación del Proceso de Evaluación:** Con las actividades del cronograma vencidas, se encuentra en condiciones para cerrar el proceso de evaluación. Para ello es importante realizar una comprobación de los datos recolectados, con el fin de estar seguros que se encuentran todos los elementos para cuantificar la evaluación.

Se contara con **RF13 Ayuda:** que permitirá brindar mayor facilidad de empleo de la aplicación a los usuarios ya autenticados.

Permitirá **RF14 Autenticar usuario:** De acuerdo con la correcta autenticación la aplicación tomara la vista de evaluación o de administración

## Capítulo 2: Características del Sistema

Pretendemos crear un modulo de administración con la idea de **RF15 Gestionar un usuario**, **RF18 Gestionar Perfil o Rol** y **RF21 Gestionar Permiso**, con la idea que usuario tenga un perfil y este una serie de permisos asociados.

---

### REQUERIMIENTOS NO FUNCIONALES

---

Los requerimientos no funcionales especifican propiedades o cualidades que el software debe tener, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, entre otras. Entre los requerimientos no funcionales del sistema propuesto se encuentran: (Pressman, 1999)

1. **Usabilidad:** La interfaz del sistema a desarrollar se debe ajustar a la estructura de los Portales de la Infraestructura Productiva, lo que posibilitará que sea familiar para los usuarios que interactuarán con el mismo. El sistema podrá ser usado de forma intuitiva por cualquier estudiante o profesor de la UCI.
2. **Apariencia o interfaz externa:** Brindar una interfaz de usuario con apariencia profesional pero a la vez amigable e intuitiva, que ofrezca una interacción efectiva al usuario.
3. **Portabilidad:** Sistema multiplataforma, permitiendo así que el Portal pueda ser visto desde cualquier sistema operativo.
4. **Rendimiento:** Debe contar con un tiempo de repuesta rápido ante las peticiones del usuario, de forma que éste se sienta cómodo navegando por el portal.
5. **Software:** Se requiere como Sistema Operativo, Windows 2000 ó superior, Linux.
6. **Hardware:** En la PC que hace función de servidor se utilizará un procesador Pentium IV con 2.4GHz, una tarjeta de red Ethernet (10/100 Mbps) y 100 MB o superior de espacio libre en el disco duro y 1GB mínimo de Memoria RAM. En las computadoras clientes se requiere un procesador Pentium con al menos 500 MHz de velocidad y conexión con la Intranet universitaria, mínimo de 512 MB de Memoria RAM.
7. **Seguridad:** La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación, para ello, debe contar con la opción de autenticación. Se deben establecer Perfiles o Roles, que garanticen que cada usuario visualice o realice solo las actividades permitidas a su rol. La información debe ser mostrada a los usuarios en el momento que la necesite, en dependencia del rol al que esté asignado. Solo el administrador puede darle algunas opciones administrativas, como son

## Capítulo 2: Características del Sistema

adicionar, modificar o eliminar elementos del sistema a determinados usuarios, en determinados momentos. La computadora que tendrá instalado el Software debe encontrarse en el área del Departamento de Pruebas, a la que sólo tiene acceso, personal autorizado.

**8. Soporte:** Se crearán privilegios en el portal, para que un personal calificado definido en el Departamento de Pruebas, actualice constantemente la información. El sistema debe ser construido de forma que permita manipular la información fácilmente y adicionar, eliminar o modificar una funcionalidad que se requiera en un momento determinado por parte de los administradores del sistema.

### DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

---

#### ACTORES DEL SISTEMA

---

Actor	Justificación
Administrador	Usuarios con permisos de administración del sitio
Director	Usuarios que tendrán de permisos de establecer el nivel de conformidad y realizar ranking por solicitud.
Especialista	Usuario con permisos para gestionar todas las funcionalidades que desencadenan en la ejecución final de la evaluación
Auditor	Usuario con permisos de realizar comprobaciones al proceso de evaluación

**Tabla 3.** Actores del Sistema

---

### DIAGRAMA DE CASO DE USO DEL SISTEMA

---

## Capítulo 2: Características del Sistema

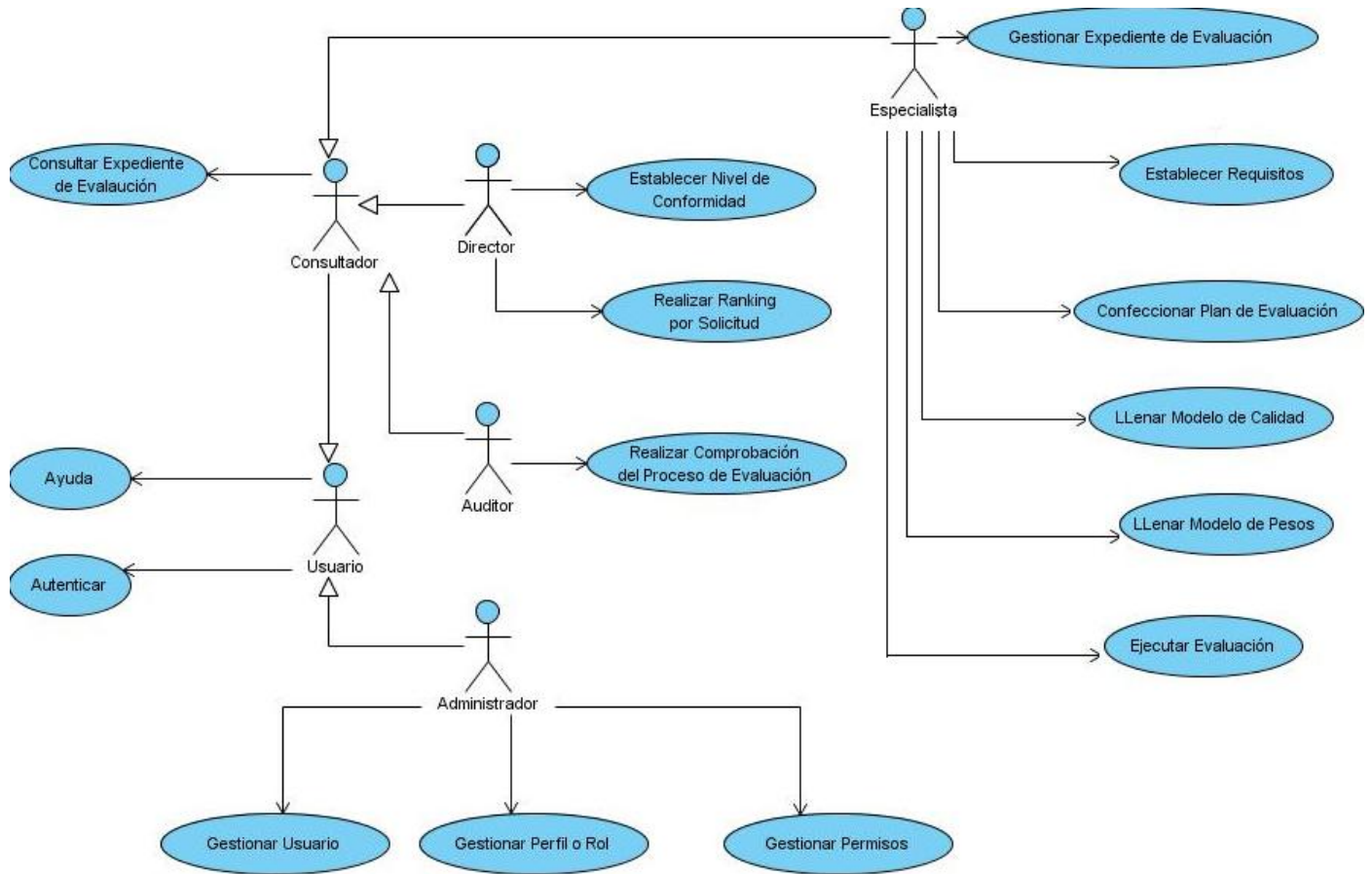


Figura 4. Diagrama de Casos de Uso del Sistema

### DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA

<b>CU-1</b>	<b>Gestionar Expediente de Evaluación</b>
<b>Propósito</b>	Cada producto- proyecto tendrá un Expediente con toda la información referente a él.
<b>Actores</b>	Especialista.
<b>Resumen</b>	El caso de uso inicia cuando el Especialista accede al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. El especialista es libre de

## Capítulo 2: Características del Sistema

	editar el contenido de los expedientes.
<b>Referencias</b>	RF1
<b>Precondiciones</b>	Debe existir un usuario autenticado como especialista.
<b>Poscondiciones</b>	Se crea el expediente de Evaluación
<b>Flujo Normal de Eventos</b>	
<b>Escenario: Crear Expediente para la Evaluación</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el especialista escoge la opción de crear Expediente para la Evaluación.</p> <p>3. El especialista escoge la opción Nuevo Expediente.</p> <p>5. El especialista llena los campos y pulsa Crear.</p>	<p>2. El sistema muestra el listado de expedientes existentes.</p> <p>4. El sistema muestra los campos a llenar</p> <ul style="list-style-type: none"> <li>• Datos</li> <li>• Funcionalidad</li> <li>• Usabilidad</li> <li>• Eficiencia</li> <li>• Mantenibilidad</li> <li>• Portabilidad</li> <li>• Pesos</li> <li>• Plan</li> <li>• Requisitos</li> </ul> <p>6. El caso de uso termina cuando el sistema guarda los cambios realizados y crea el Expediente.</p>
<b>Escenario: Modificar Expediente para la Evaluación.</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el especialista escoge la opción de crear Expediente para la Evaluación.</p> <p>3. El especialista escoge el Expediente.</p>	<p>2. El sistema muestra un listado todas los Expediente existentes en la aplicación.</p> <p>4. El sistema muestra todos los datos del expediente.</p>



## Capítulo 2: Características del Sistema

5. El especialista selecciona el campo y pulsa editar. 7. El especialista confirma.	6. El sistema pide confirmación del deseo de modificación. 8. El caso de uso termina cuando el sistema guarda los cambios realizados.
<b>Escenario: Eliminar Expediente para la Evaluación.</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el especialista escoge la opción de crear Expediente para la Evaluación. 3. El especialista escoge el Expediente. 5. El especialista selecciona el campo y pulsa Eliminar. 7. El especialista Aceptar.	2. El sistema muestra un listado todas los Expediente existentes en la aplicación. 4. El sistema muestra todos los datos del expediente. 6. El sistema pide confirmación del deseo de eliminar. 8. El caso de uso termina cuando el sistema guarda los cambios realizados.
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario no esta autenticado.	2. El sistema muestra mensaje de acceso restringido.

**Tabla 4.** Descripción extendida CU-1 Gestionar Expediente.

<b>CU-2</b>	<b>Consultar Expediente de Evaluación</b>
Propósito	El expediente será consultado por el personal autorizado.
Actores	Consultador
Resumen	El caso de uso inicia cuando el consultador accede al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. Solo entonces podrán consultar el Expediente de Evaluación.
Referencias	RF4

## Capítulo 2: Características del Sistema

Precondiciones	Debe existir un usuario autenticado como especialista.
Poscondiciones	Muestra el expediente para se consultado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el consultador escoge la opción de Crear Expediente para la Evaluación. 3. El consultador escoge la Expediente que desea consultar.	2. El sistema muestra un listado todas las Expediente existentes en la aplicación. 4. El caso de uso termina cuando el sistema muestra el Expediente en su totalidad solo a manera de consulta.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 5.** Descripción extendida CU-2 Consultar Expediente.

<b>CU-3</b>	<b>Establecer Requisitos</b>
Propósito	Guiará el proceso de evaluación.
Actores	Especialista
Resumen	El caso de uso inicia cuando el especialista acceda al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. El especialista es libre de establecer los requisitos para la evaluación.
Referencias	RF5
Precondiciones	Debe existir un usuario autenticado como especialista.
Poscondiciones	Se establecen los requisitos para la evaluación
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el especialista escoge la opción Requisitos Establecidos 3. El especialista escoge la opción Nuevos	2. El sistema muestra los requisitos establecidos 4. El sistema muestra los campos a llenar

## Capítulo 2: Características del Sistema

Requisitos	para Crear la Planilla de Requisitos:
5. El especialista llena los campos y pulsa guardar.	<ul style="list-style-type: none"> <li>• Identificador</li> <li>• Requisitos Establecidos</li> <li>• Pre-requisitos del Cliente</li> <li>• Pre-Requisitos del Evaluador</li> </ul>
	6. El caso de uso termina cuando el sistema guarda el estado de los campos.
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El especialista no completo los campos.	2. Señala los campos obligatorios.
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El especialista no esta autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 6.** Descripción extendida CU-3 Establecer Requisitos.

<b>CU-4</b>	<b>Confeccionar Plan de Evaluación</b>
Propósito	Prevé tener concretamente toda una serie de acciones específicas que tributen a lo ya establecido recogidas en un mismo sitio.
Actores	Especialista
Resumen	El caso de uso inicia cuando el especialista acceda al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. El especialista es libre de confeccionar el Plan de Evaluación.
Referencias	RF6
Precondiciones	Debe existir un usuario autenticado como especialista.
Poscondiciones	Se conforma el plan de evaluación.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

## Capítulo 2: Características del Sistema

1. El caso de uso se inicia cuando el especialista escoge la opción Plan de Evaluación.	2. El sistema muestra el listado de todos los planes existentes.
3. El especialista pulsa Nuevo Plan de Evaluación	4. El sistema muestra los campos a llenar para Confeccionar el Plan:
5. El especialista llena los campos y pulsa guardar.	<ul style="list-style-type: none"> <li>• Identificador</li> <li>• Métricas</li> <li>• Cronograma de Actividades</li> </ul>
	6. El caso de uso termina cuando el sistema guarda el estado de los campos.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El especialista no completo los campos.	2. Señala los campos obligatorios.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 7.** Descripción extendida CU-4 Confeccionar Plan de Evaluación.

CU-5	Llenar Modelo de Calidad
Propósito	Permite establecer cuantitativamente la incidencia de una cualidad determinada en el producto evaluado.
Actores	Especialista
Resumen	El caso de uso inicia cuando el Especialista acceder al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. El especialista es libre de editar contenido del modelo de calidad.
Referencias	RF7
Precondiciones	Debe existir un usuario autenticado como especialista.
Poscondiciones	Se establece el Modelo de Calidad.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

## Capítulo 2: Características del Sistema

<p>1. El caso de uso se inicia cuando el especialista escoge la opción Modelo de Calidad.</p> <p>3. El especialista selecciona las características Funcionalidad.</p> <p>5. El especialista llena los campos de cada sub característica y pulsa guardar.</p>	<p>2. El sistema muestra las características a llenar:</p> <ul style="list-style-type: none"> <li>• Funcionalidad:</li> <li>• Fiabilidad:</li> <li>• Usabilidad:</li> <li>• Eficiencia:</li> <li>• Mantenibilidad:</li> </ul> <p>4. El sistema muestra las sub características a llenar por características seleccionada:</p> <p>Funcionalidad:</p> <ul style="list-style-type: none"> <li>• Identificador</li> <li>• Idoneidad</li> <li>• Exactitud</li> <li>• Interoperabilidad</li> <li>• Seguridad</li> <li>• Cumplimiento a Normas</li> </ul> <p>6. El sistema guarda el estado de los campos.</p>
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El especialista selecciona las características Fiabilidad.</p> <p>3. El especialista llena los campos de cada sub característica y pulsa guardar.</p>	<p>2.El sistema muestra las sub características a llenar por características seleccionada:</p> <ul style="list-style-type: none"> <li>• Identificador</li> <li>• Madurez</li> <li>• Recuperabilidad</li> <li>• Tolerancia a Fallos</li> </ul> <p>4. El sistema guarda el estado de los campos.</p>
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>

## Capítulo 2: Características del Sistema

1. El especialista selecciona las características Eficiencia. 3. El especialista llena los campos de cada sub característica y pulsa guardar.	2. El sistema muestra las sub características a llenar por características seleccionada: <ul style="list-style-type: none"> <li>• Identificador</li> <li>• Comportamiento frente al Tiempo</li> <li>• Comportamiento de Recursos</li> </ul> 4. El sistema guarda el estado de los campos.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El especialista selecciona las características Usabilidad. 3. El especialista llena los campos de cada sub característica y pulsa guardar.	2.El sistema muestra las sub características a llenar por características seleccionada: <ul style="list-style-type: none"> <li>• Identificador</li> <li>• Aprendizaje</li> <li>• Comprensión</li> <li>• Operatividad</li> <li>• Atractividad</li> </ul> 4. El sistema guarda el estado de los campos.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El especialista selecciona las características Mantenibilidad. 3. El especialista llena los campos de cada sub característica y pulsa guardar.	2.El sistema muestra las sub características a llenar por características seleccionada: <ul style="list-style-type: none"> <li>• Identificador</li> <li>• Estabilidad</li> <li>• Facilidad de Análisis</li> <li>• Facilidad de Cambio</li> <li>• Facilidad de Pruebas</li> </ul> 4. El sistema guarda el estado de los campos.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El especialista selecciona las características Portabilidad.	2. El sistema muestra las sub características a llenar por características seleccionada:

## Capítulo 2: Características del Sistema

3. El especialista llena los campos de cada sub característica y pulsa guardar.	<ul style="list-style-type: none"> <li>• Capacidad de Instalación</li> <li>• Capacidad de Emplazamiento</li> <li>• Adaptabilidad</li> <li>• Coexistencia</li> </ul> <p>4. El sistema guarda el estado de los campos.</p>
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 8.** Descripción extendida CU-5 Llenar Modelo de Calidad.

CU-6	Llenar Modelo de Pesos
Propósito	Provee un modelo general definiendo los valores que tomaran las características y sub-características por las que se obtendrá la evaluación.
Actores	Especialista.
Resumen	El caso de uso inicia cuando el Especialista acceder al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. El especialista es libre de editar contenido de los modelos de pesos.
Referencias	RF8
Precondiciones	Debe existir un usuario autenticado como especialista.
Poscondiciones	Quedará conformado el Modelo de Pesos para la Evaluación.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el especialista escoge la opción Modelo de peso.	2. El sistema muestra el listado de Modelo de Pesos
3. El especialista selecciona Nuevo Modelo de peso.	4. El sistema muestra los campos a completar para Llenar el Modelo de Pesos:
5. El especialista llena los campos y pulsa	1. Funcionalidad

## Capítulo 2: Características del Sistema

guardar	<ul style="list-style-type: none"><li>1.1 idoneidad</li><li>1.2 exactitud</li><li>1.3 interoperabilidad</li><li>1.4 seguridad</li><li>1.5 cumplimientoDeNormas</li><li><b>2. Fiabilidad</b><ul style="list-style-type: none"><li>2.1 madurez</li><li>2.2 recuperabilidad</li><li>2.3 toleranciaAFallos</li></ul></li><li><b>3. Usabilidad</b><ul style="list-style-type: none"><li>3.1 aprendizaje</li><li>3.2 comprensión</li><li>3.3 operatividad</li><li>3.4 atraktividad</li></ul></li><li><b>4. Eficiencia</b><ul style="list-style-type: none"><li>4.1 comportaminetoEnEITiempo</li><li>4.2 comportamientoDeRecursos</li></ul></li><li><b>5. Mantenibilidad</b><ul style="list-style-type: none"><li>5.1 estabilidad</li><li>5.2 facilidadDeAnalisis</li><li>5.3 facilidadDeCambio</li><li>5.4 facilidadDePruebas</li></ul></li><li><b>6. Portabilidad</b><ul style="list-style-type: none"><li>6.1 capacidadDeInstalacion</li><li>6.2 capacidadDereemplazamiento</li><li>6.3 adaptabilidad</li><li>6.4 coexistencia</li></ul></li></ul> <p>6. El caso de uso termina cuando el sistema guarda el estado de los campos.</p>
---------	---



## Capítulo 2: Características del Sistema

Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 9.** Descripción extendida CU-6 Llenar Modelo de Pesos.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el director escoge la opción Establecer Nivel de Aceptación. 3. El director puede modificar el valor de los niveles existentes de conformidad.	2. El sistema muestra la vista de la opción deseada para completar los datos. <ul style="list-style-type: none"> <li>• Menor Valor para rango inaceptado</li> <li>• Mayor Valor para rango inaceptado</li> <li>• Menor Valor para rango mínimo aceptado</li> <li>• Mayor Valor para rango mínimo aceptado</li> <li>• Menor Valor para rango Aceptado</li> <li>• Mayor Valor para rango Aceptado</li> </ul> 6. El caso de uso termina cuando el sistema

## Capítulo 2: Características del Sistema

	guarda los niveles establecidos por el director.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El director no completo los campos.	2. Señala los campos obligatorios.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El director entro información no valida.	2. Muestra los campos no validos subrayados.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 10.** Descripción extendida CU-7 Establecer Nivel de Conformidad.

CU-8	Ejecutar Evaluación
Propósito	Obtener una evaluación cuantitativa del producto software.
Actores	Especialista.
Resumen	El caso de uso inicia cuando el especialista acceda al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. El especialista tendrá permisos para ejecutar la Evaluación.
Referencias	RF10
Precondiciones	Debe existir un usuario autenticado como especialista.
Poscondiciones	El producto quedará evaluado cuantitativamente.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el especialista pulsa Crear Expediente. 3. El especialista llena los campos y pulsa crear.	2. El sistema muestra el listado de características y sub características para ser asociadas con los datos entrados 4. El caso de uso termina cuando el sistema al crear el expediente calcula automáticamente el valor de la evaluación.

## Capítulo 2: Características del Sistema

Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El especialista no completo los campos.	2. Señala los campos obligatorios.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El especialista entró números no validos.	2. Muestra mensaje de error.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 11.** Descripción extendida CU-8 Ejecutar Evaluación.

CU-9	Realizar Ranking por Solicitud
Propósito	Permitirá tener una visión del nivel de competencia de acuerdo a parámetros determinado.
Actores	Director
Resumen	El caso de uso inicia cuando el Director accede al módulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. Solo entonces podrá establecer el Ranking por Solicitud.
Referencias	RF11
Precondiciones	Debe existir un usuario autenticado como director.
Poscondiciones	Mostrará la ubicación del producto dentro de una lista de productos evaluados
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
3. El caso de uso se inicia cuando el director escoge la opción Realizar Ranking. 5. El director escoge el campo por el que desea establecer el Ranking.	2. El sistema muestra una tabla con todos los productos evaluados con las evaluaciones por características y general del mismo. 6. El caso de uso termina cuando el sistema muestra el ranking por la selección del director.
Flujo Alterno	

## Capítulo 2: Características del Sistema

Acción del actor	Respuesta del Sistema
1. El director no escogió el campo por el que desea hacer la solicitud.	2. El sistema no muestra variaciones en el estado del resto de los campos.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 12.** Descripción extendida CU-9 Realizar Ranking por Solicitud.

<b>CU-10</b>	<b>Realizar comprobación del proceso de evaluación</b>
<b>Propósito</b>	Permitirá comprobar la calidad del proceso de evaluación.
<b>Actores</b>	Auditor.
<b>Resumen</b>	El caso de uso inicia cuando el Auditor accede al modulo de evaluación, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo evaluación. Solo entonces podrá establecer la comprobación del proceso de evaluación.
<b>Referencias</b>	RF12
<b>Precondiciones</b>	Debe existir un usuario autenticado como Auditor.
<b>Poscondiciones</b>	La Evaluación quedará comprobada
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el auditor escoge la opción Realizar comprobación del proceso de evaluación. 3. El auditor escoge el que desee. 5. El auditor llena los campos y pulsa guardar la comprobación.	2. El sistema muestra el listado de expedientes disponibles para realizar la comprobación. 4. El sistema muestra los campos que debe llenar para que la comprobación sea válida. <ul style="list-style-type: none"> <li>• Auditor</li> <li>• Resultado del Proceso</li> <li>• Comentario</li> <li>• Expediente expediente</li> </ul> 6. El caso de uso termina cuando el sistema

## Capítulo 2: Características del Sistema

	guarda la comprobación.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El auditor no selecciono ningún expediente.	2. El sistema no muestra campos a llenar.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El director entro información no válida.	2. Muestra mensaje de error.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario no está autenticado.	2. Muestra mensaje de acceso restringido.

**Tabla 13.** Descripción extendida CU-10 Realizar Comprobación del Proceso de Evaluación.

<b>CU-11</b>	<b>Ayuda</b>
Propósito	Brindará asesoramiento sobre los aspectos más importantes del sitio.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el Usuario accede a la página principal, en este momento el sistema le da acceso a consultar la ayuda.
Referencias	RF13
Precondiciones	Debe existir un usuario con permiso de entrar al portal.
Poscondiciones	Consultará los temas relacionados con el proceso de evaluación.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el usuario accede a la ayuda.	2. El caso de uso termina cuando el sistema le muestra la ayuda.
Flujo Alterno	
Acción del actor	Respuesta del Sistema
1. El usuario intenta hacer otra operación.	2. Muestra mensaje de acceso restringido.

**Tabla 14.** Descripción extendida CU-11 Ayuda.

<b>CU-12</b>	<b>Autenticar Usuario</b>
Propósito	Controlar acceso a zonas restringidas.
Resumen	Un usuario con privilegios de autenticación accede a la página para autenticarse.

## Capítulo 2: Características del Sistema

Referencias	RF14
Precondiciones	Debe de ser posible acceder a la aplicación
Poscondiciones	El Usuario queda autenticado.
<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 entra a la página de autenticación.	2. El caso de uso termina cuando el usuario queda autenticado.
<b>Flujo Alterno</b>	
Acción del actor	Respuesta del Sistema
1. El usuario no existe en la aplicación.	1. Muestra mensaje de acceso restringido.

**Tabla 15.** Descripción extendida CU-12 Autenticar Usuario.

<b>CU-13</b>	<b>Gestionar Usuario.</b>
Propósito	Gestionar el estado de un usuario.
Actores	Administrador.
Resumen	El caso de uso inicia cuando el administrador acceder al módulo de administración, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo administración. El administrador es libre de editar contenido de usuarios.
Referencias	RF15, RF16, RF17
Precondiciones	Debe existir un usuario autenticado con permisos para administrar el portal.
Poscondiciones	Quedan usuarios creados, modificados o eliminados
<b>Flujo Normal de Eventos</b>	
<b>Escenario: Insertar Usuario.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el administrador escoge la opción de insertar usuario.  3. El administrador llena los datos del nuevo usuario.	2. El sistema muestra el formulario correspondiente a la opción seleccionada. <ul style="list-style-type: none"> <li>• Login</li> <li>• Nombre Completo</li> <li>• Password</li> <li>• Habilitado</li> <li>• Descripción</li> </ul>

## Capítulo 2: Características del Sistema

	<ul style="list-style-type: none"> <li>• Email</li> <li>• Mostrar Email</li> <li>• Asignar Perfil Perfil_Admin</li> </ul> <p>4. El caso de uso termina cuando el sistema guarda los cambios realizados.</p>
<b>Escenario: Modificar Usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el administrador escoge la opción de modificar usuario.</p> <p>3. El administrador selecciona el que desea modificar.</p> <p>5. El administrador modifica los campos deseados.</p>	<p>2. El sistema muestra el listado de usuarios registrados en la aplicación.</p> <p>4. El sistema muestra los datos de la persona en cuestión.</p> <p>6. El caso de uso termina cuando el sistema guarda los cambios realizados.</p>
<b>Escenario: Eliminar Usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El caso de uso se inicia cuando el administrador escoge la opción de eliminar usuario.</p> <p>3. El administrador selecciona el que desea eliminar.</p> <p>5. El administrador Acepta.</p>	<p>2. El sistema muestra listado de usuarios registrados en la aplicación.</p> <p>4. El sistema muestra un mensaje de “Esta Seguro que desea eliminar este Usuario”.</p> <p>6. El caso de uso termina cuando el sistema guarda los cambios realizados.</p>
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario intenta modificar el identificador.	2. Muestra mensaje de error.
<b>Flujo Alterno</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario no está autenticado	2. Muestra mensaje de acceso restringido

**Tabla 16.** Descripción extendida CU-13 Gestionar Usuario.

<b>CU-14</b>	<b>Gestionar Perfil o Rol</b>
--------------	-------------------------------

## Capítulo 2: Características del Sistema

Propósito	Gestiona el Perfil que puede tener un usuario.
Actores	Administrador.
Resumen	El caso de uso inicia cuando el administrador accede al módulo de administración, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo administración. El administrador es libre de editar contenido de Perfiles o Roles.
Referencias	RF18, RF19, RF20
Precondiciones	Debe existir un usuario autenticado con permisos para administrar el portal.
Poscondiciones	Queda creado, modificado o eliminado un Perfil
<b>Flujo Normal de Eventos</b>	
<b>Escenario: Insertar Perfil o Rol</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando el administrador escoge la opción de insertar Perfil.</li> <li>3. El administrador llena los campos.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema muestra los campos a llenar: <ul style="list-style-type: none"> <li>• Nombre del Perfil</li> <li>• Descripción</li> </ul> </li> <li>4. El caso de uso termina cuando el sistema guarda los cambios realizados.</li> </ol>
<b>Escenario: Modificar Perfil o Rol</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando el administrador escoge la opción de modificar Perfil.</li> <li>3. El administrador escoge el Perfil.</li> <li>5. El administrador modifica los campos deseados.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema muestra un listado todos los Perfiles existentes en la aplicación.</li> <li>4. El sistema muestra los datos del Perfil en cuestión.</li> <li>6. El caso de uso termina cuando el sistema guarda los cambios realizados.</li> </ol>
<b>Escenario: Eliminar Perfil o Rol</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando el administrador escoge la opción de eliminar</li> </ol>	<ol style="list-style-type: none"> <li>2. El caso de uso termina cuando el sistema muestra un listado de los Perfiles existentes</li> </ol>



## Capítulo 2: Características del Sistema

Perfil. 3. El administrador escoge el Perfil. 5. El administrador confirma.	en la aplicación. 4. El sistema muestra un mensaje de “Esta Seguro que desea eliminar el Perfil”. 6. El sistema guarda los cambios realizados.
---	--

**Tabla 17.** Descripción extendida CU-13 Gestionar Perfil.

<b>CU-15</b>	<b>Gestionar Permiso</b>
Propósito	Gestiona los permisos que puede tener un usuario.
Actores	Administrador
Resumen	El caso de uso inicia cuando el administrador accede al módulo de administración, en este momento el sistema chequea que el usuario tenga permisos para la operación que solicita, si los tiene, el sistema muestra el portal en modo administración. El administrador es libre de gestionar los permisos de la aplicación.
Referencias	RF21, RF22, RF23
Precondiciones	Debe existir un usuario autenticado con permisos para administrar el portal.
Poscondiciones	Queda creado, modificado o eliminado un Permiso.
<b>Flujo Normal de Eventos</b>	
<b>Escenario: Insertar Permisos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el administrador escoge la opción de insertar Permisos. 3. El administrador llena los el campo.	2. El sistema muestra los campos a llenar: <ul style="list-style-type: none"> <li>• URL Pattern</li> <li>• Roles (separados por(,))</li> </ul> 4. El caso de uso termina cuando el sistema guarda los cambios realizados.
<b>Escenario: Modificar Permiso</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el administrador escoge la opción de modificar Permisos.	2. El sistema muestra un listado todos los Permisos existentes en la aplicación. 4. El sistema pide confirmación.

## Capítulo 2: Características del Sistema

3. El administrador modifica el campo Perfil con el que se relaciona. 5. El administrador confirma.	6. El caso de uso termina cuando el sistema guarda los cambios realizados.
<b>Escenario: Eliminar Permiso</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el administrador escoge la opción de eliminar Permiso. 3. El administrador escoge el Permiso. 5. El administrador Acepta.	2. El sistema muestra el listado de Permisos registrados en la aplicación. 4. El sistema muestra un mensaje de “Esta Seguro que desea eliminar el Permiso”. 6. El caso de uso termina cuando el sistema guarda los cambios realizados.

**Tabla 18.** Descripción extendida CU-15 Gestionar Permisos.

---

### CONCLUSIONES PARCIALES

---

Con el desarrollo de este capítulo se obtuvo una mejor comprensión del problema en cuestión, mediante el modelado del dominio, la definición de los requerimientos funcionales y no funcionales y la descripción del sistema propuesto, obteniéndose las funciones que debe tener el portal.

En este capítulo se comenzó a desarrollar la propuesta de solución, determinando un listado con las propiedades que debe cumplir el sistema y las funcionalidades que debe realizar, representando las últimas mediante un diagrama de casos de uso del sistema y finalmente describiendo paso a paso todas las acciones del actor con los casos de uso que interactúa. Ahora se puede comenzar a construir el sistema, cumpliendo con todos los requerimientos y las funciones que se han considerado necesarias en este capítulo.

---

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA**

---

### **INTRODUCCIÓN**

---

Para el desarrollo de un software es necesario contar con descripciones detalladas y saber cómo se satisfacen los requerimientos y las restricciones. En este capítulo se describen los elementos más importantes correspondientes a la etapa de análisis y diseño del sistema, para eso se definen los diagramas de interacción (secuencia) correspondientes a cada caso de uso, detallados por escenario para lograr su mayor comprensión y los diagramas de clases de análisis y diseño del sistema (Wesley, 2005)

### **MODELADO DE ANÁLISIS**

---

El análisis se basa en un modelo de objetos conceptuales, que se denomina modelo de análisis. Este modelo estructura los requisitos de un modo que facilita su comprensión, su modificación, en general, su mantenimiento. Se describe utilizando el lenguaje de los desarrolladores, por tanto puede introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema. (Larman, 1999)

Durante el análisis se analizan los requisitos que se obtuvieron en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura. Siempre hay que tener presente que es más fácil y menos costoso hacer cambios o arreglar defectos en el análisis que en las fases siguientes.

---

### **DIAGRAMA DE CLASES DEL ANÁLISIS**

---

El diagrama de clases del análisis representa las clases del análisis que participan en las realizaciones de los casos de uso y las relaciones que se establecen entre ellas. Un diagrama de clases del análisis representa las cosas del mundo real y no de la implementación, se define como aquel artefacto en el que se figuran los conceptos en un dominio del problema. Una clase de análisis representa una abstracción de

## Capítulo 3: Análisis y Diseño del Sistema

una o varias clases y/o subsistemas del diseño del sistema. Esta abstracción se centra en el tratamiento de los requisitos funcionales. (Quatrani, Terry, 2002)

Ver parte de los diagramas de clases de Análisis en el [Anexo1](#)

---

### ENTIDADES DE CLASES DE ANÁLISIS

---

A continuación una breve explicación de las entidades con las que se va a trabajar en los diagramas de clases de análisis

Entidad	Descripción
Expediente de Evaluación	En ella se almacenan los expedientes de evaluación que se gestionan.
Requisitos	Se guardan los requisitos por los que se va a realizar la evaluación.
Modelo de Pesos	Se guardan los modelos de pesos que se van a utilizar para realizar la evaluación.
Evaluación	Es donde se guarda los detalles de la evaluación que se va a realizar.
Plan de Evaluación	Se guarda lo referente al plan de evaluación que se va a aplicar a la evaluación.
Conformidad	Permitirá guardar una evaluación cualitativa del producto.
Ranking por Solicitud	Se guarda el conjunto parámetros para establecer el Ranking.
Ayuda	Se guarda el contenido de ayuda que es publicado en el portal.
Usuario	En ella se almacena el personal que interactúa con el Portal.
Perfil o Rol	En ella se guardan los perfiles que existen en el Portal.
Permisos	En ella se guardan los perfiles que existen en el Portal.

Tabla 19. Descripción de las Entidades del Sistema.

---

### ARQUITECTURA DE SOFTWARE

---

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación y tiene la responsabilidad de:

- Definir los módulos principales
- Definir las responsabilidades que tendrá cada uno de estos módulos

# Capítulo 3: Análisis y Diseño del Sistema

- Definir la interacción que existirá entre dichos módulos
- Control y flujo de datos (Casanovas, 2005)

---

## DESCRIPCIÓN DE LA ARQUITECTURA UTILIZADA

---

Como la mayoría de los framework de desarrollo web, Grails está basado en el patrón Modelo Vista Controlador (MVC). En Grails los modelos son tratados como clases de dominio que permiten a la aplicación mostrar los datos en la vista. A diferencia de en otros framework, las clases de dominio de Grails son automáticamente persistidas y es incluso posible generar el esquema de la base de datos. Los controladores por su parte, permiten gestionar las peticiones a la aplicación y organizar los servicios proporcionados. Por último, la vista por defecto en Grails son las Groovy Server Pages (GSP) y habitualmente nos muestra el contenido en formato HTML. (Graeme, et al., 2009)

---

## DEFINICIONES DE DISEÑO QUE SE APLICAN

---

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser rehusado, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Los patrones de diseño pretenden: (LARMAN, C. 1999.).

No ser reiterativos en la búsqueda de soluciones a problemas que son conocidos y ya han sido solucionados.

- Construir un vocabulario que sea común entre los diseñadores, para de esta manera estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.
- Proporcionar catálogos de elementos rehusados en el diseño de sistemas software.

# Capítulo 3: Análisis y Diseño del Sistema

Los patrones GRASP<sup>6</sup> describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Los primeros cinco patrones GRASP son:

- Experto: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- Creador: Asignarle a la clase B la responsabilidad de crear una instancia de clase A
- Alta cohesión: Asignar una responsabilidad de modo que la cohesión siga siendo alta.
- Bajo Acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento.
- Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase (REYNOSO, 2005)

## DISEÑO

---

En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación. Durante la fase de construcción, cuando la arquitectura es estable y los requisitos están bien entendidos, el centro de atención se desplaza a la implementación.

---

## DIAGRAMA DE INTERACCIÓN

---

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Estos están compuestos por un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia. (Quatrani, Terry, 2002)

---

<sup>6</sup> GRASP (General Responsibility Assignment Software Patterns).

# Capítulo 3: Análisis y Diseño del Sistema

---

## DIAGRAMA DE SECUENCIA DEL DISEÑO

---

En el diseño, es recomendable representar las interacciones de los objetos que participan en la realización de un caso de uso, mediante diagramas de secuencias de esta forma podemos encontrar mejor las secuencias de interacciones detalladas y ordenadas en el tiempo. En los diagramas de secuencias, se muestran las interacciones entre objetos mediante transferencias de mensajes entre objetos o subsistemas. (Larman, 1999)

**Ver parte de los diagramas de secuencia del diseño por caso de uso en el [Anexo2](#)**

---

## DIAGRAMA DE CLASES DEL DISEÑO

---

Una clase de diseño es una abstracción sin costuras de una clase o construcción similar en la implementación del sistema. (Larman, 1999)

El lenguaje utilizado para especificar una clase de diseño es lo mismo que el lenguaje de programación. Consecuentemente las operaciones, parámetros, atributos, tipos y demás son especificados utilizando la sintaxis del lenguaje de programación elegido. La visibilidad de los atributos y las operaciones de una clase de diseño se especifica con frecuencia. Las relaciones de aquellas clases de diseño implicadas con otras clases, a menudo tienen un significado directo cuando la clase es implementada. Los métodos tienen correspondencia directa con el correspondiente método en la implementación de las clases.

**Ver parte de los diagramas de clases del diseño en el [Anexo3](#)**

---

## DISEÑO DE BASE DE DATOS

---

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos.

Si las tablas no son definidas apropiadamente, podemos tener muchos problemas al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de información. No importa si nuestra base de datos tiene muchos, o pocos registros, es importante asegurarnos que nuestra base de datos está correctamente diseñada para que tenga eficiencia y que se pueda seguir utilizando por largo del tiempo. (Graeme, et al., 2009)

Grails incluye un potente framework de mapeo objeto relacional llamado Grails Objeto Mapeo Relacional (GORM). Como la mayoría de los framework Mapeo Objeto-Relacional (ORM), GORM puede mapear

## Capítulo 3: Análisis y Diseño del Sistema

objetos a bases de datos relacionales y representar las relaciones entre estos objetos, como uno a uno o uno a muchos. Pero lo que distingue a GORM, aparte de otros ORMs es el hecho de que es construido por un lenguaje dinámico como Groovy. Incluye los métodos CRUD en la clase sin tener que implementarlos o heredarlos de una superclase persistente. Además, es capaz de proporcionar un ORM DSL para métodos de búsqueda dinámicos y los criterios de búsqueda. (Christopher, et al., 2008)

---

### **MODELO DE DATOS**

---

A continuación mostramos una vista del Modelo de Datos de la Aplicación:



# Capítulo 3: Análisis y Diseño del Sistema

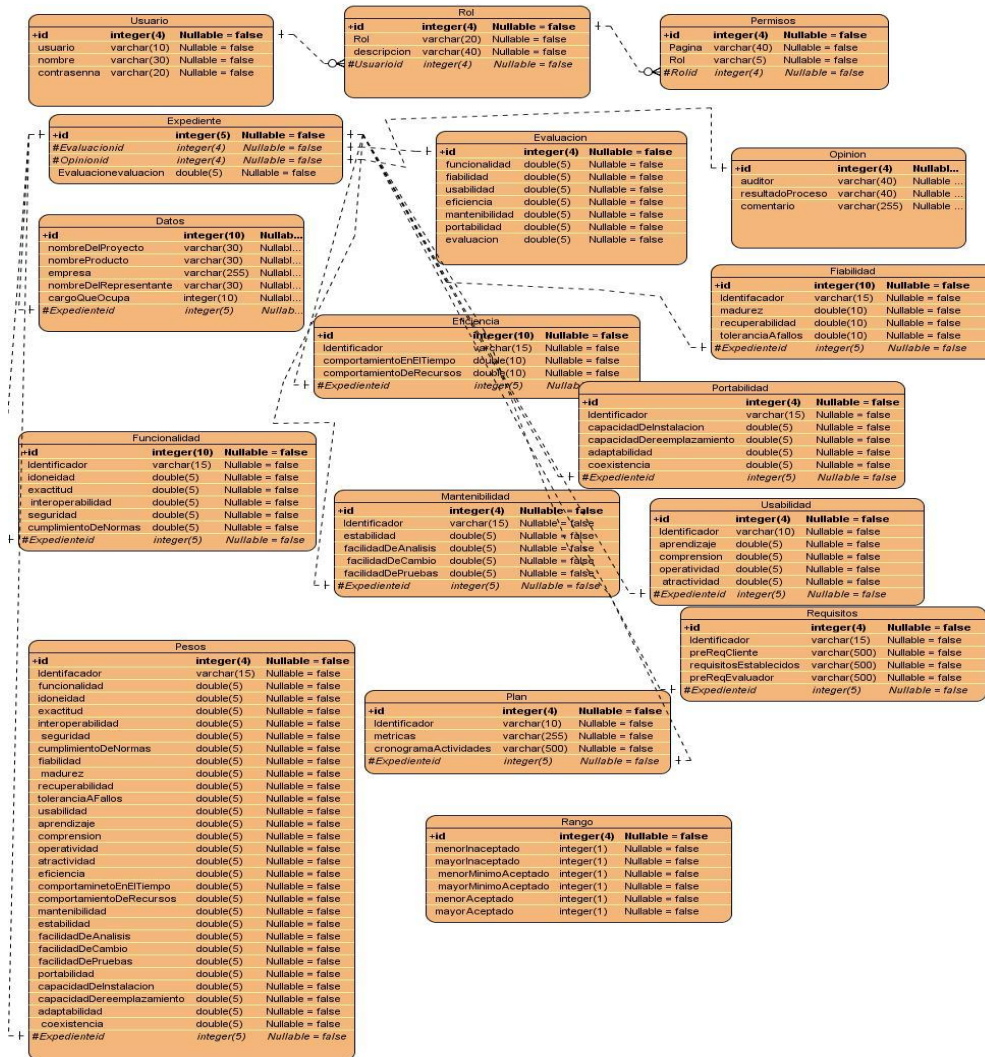


Figura 5. Modelo de Datos

## CONCLUSIONES PARCIALES

Se realizó el análisis de la propuesta de solución logrando una abstracción de la aplicación, a través de una representación de los casos de usos en un diagrama de clases del análisis. Con el modelo de diseño se logró abstraer el modelo de implementación, siendo la entrada esencial en las actividades relacionadas con la implementación, para eso se generaron diferentes artefactos como diagramas de clases del diseño, diagramas de interacción, los cuales son una aproximación real a la futura implementación.

---

## **CAPÍTULO 4: IMPLEMENTACIÓN**

---

### **INTRODUCCIÓN**

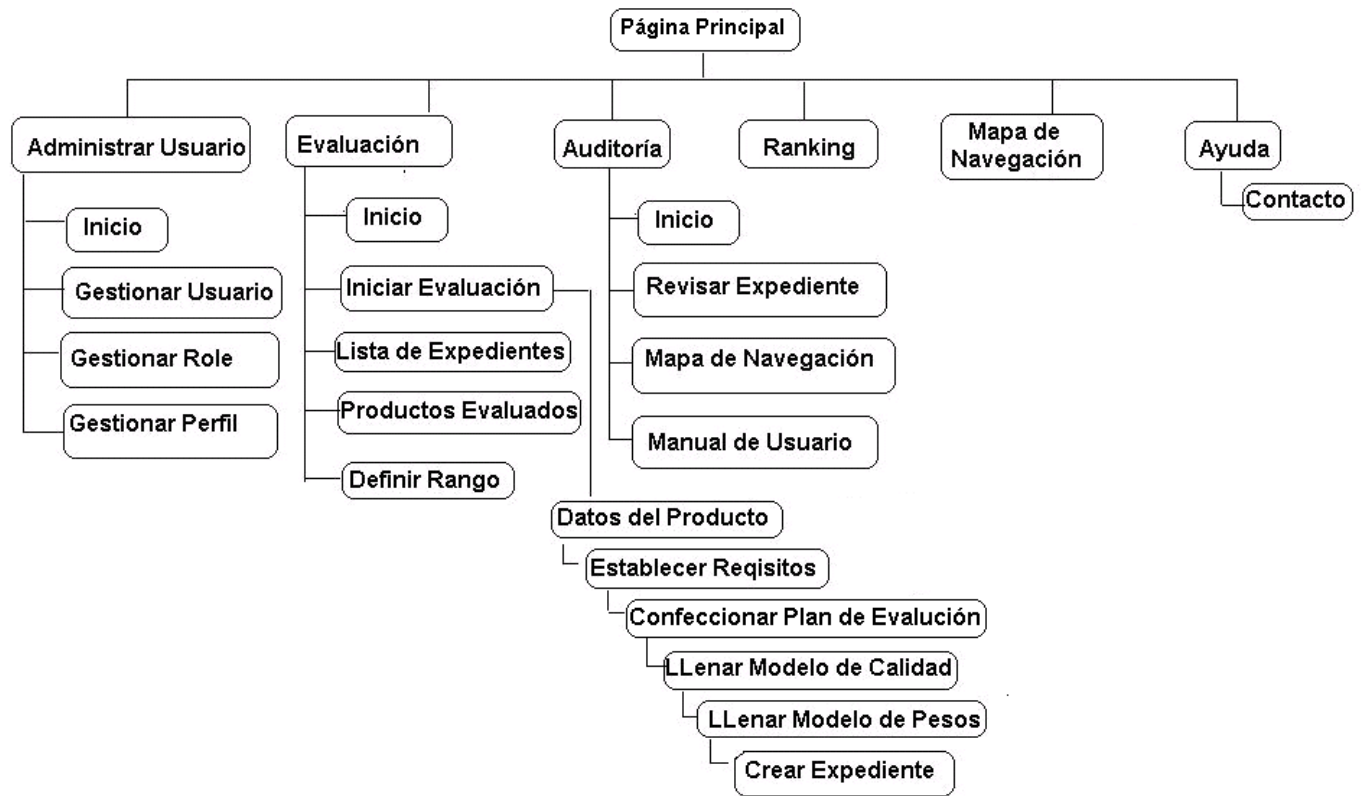
---

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se ponen en práctica todas las descripciones y arquitecturas propuestas en las fases de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso unificado de software. La implementación ofrece una materialización precisa de los requisitos. En ella se obtienen como resultado componentes de código que se compilan e integran en versiones ejecutables. (Quatrani, Terry, 2002)

### **MAPA DE NAVEGACIÓN**

---

Este es una representación visual de forma jerárquica que muestra la estructura general del sitio. A continuación se muestra el mapa conceptual del Portal



**Figura 6.** Mapa de Navegación.

## MÓDULOS

En programación un módulo es una parte de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará una de dichas tareas (o quizá varias en algún caso).

Los módulos suelen estar organizados jerárquicamente en niveles, de forma que hay un módulo superior que realiza las llamadas oportunas a los módulos del nivel inferior. Cuando un módulo es llamado, recibe como entrada los datos proporcionados por el módulo de nivel superior que ha hecho la llamada, realiza su tarea, a su vez este módulo puede llamar a otro u otros módulos de nivel inferior si fuera necesario; cuando finaliza su tarea, devuelve la salida pertinente al módulo superior que lo llamó inicialmente, y es este módulo superior el que continúa con la ejecución del programa.

En el portal se implementaron dos módulos, los que se muestran a continuación:

## Capítulo 4: Implementación

<b>Nombre:</b> Administración	<b>Paquete:</b>	<b>Dependencia:-</b>
<b>Descripción:</b> Módulo para gestionar todo lo referente a los usuarios, perfiles y permisos del portal		

**Tabla 20.** Módulo de Administración

<b>Nombre:</b> Evaluación	<b>Paquete:</b>	<b>Dependencia:-</b>
<b>Descripción:</b> Módulo para gestionar todo lo referente al proceso de evaluación		

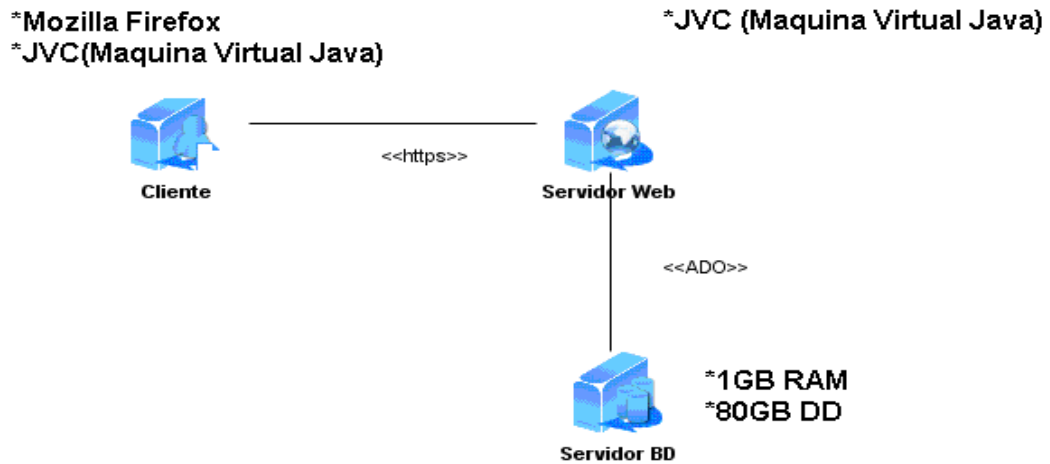
**Tabla 21.** Módulo de Evaluación

### MODELO DE DESPLIEGUE

---

El modelo de despliegue es un modelo de objeto que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos de cómputo. 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. (Larman, 1999)

El modelo de despliegue describe la distribución física de de los componentes lógicos desarrollados en la aplicación. Es decir, se sitúa el software en el hardware que lo contiene.



**Figura 7.** Diagrama de Despliegue

## CONCLUSIONES PARCIALES

---

En este capítulo se mostró la estructura del Portal luego de su implementación así como la vista del Mapa de Navegación que nos da la posibilidad de visitar el Portal sin que sea difícil la navegación por el mismo, se realizó el modelo de implementación de la aplicación, donde se obtuvieron los artefactos fundamentales propuestos por RUP, uno de ellos es el modelo de despliegue que permitió definir la arquitectura física del sistema, además de hacer una explicación de los dos módulos con los que cuenta la aplicación.

## **CONCLUSIONES GENERALES**

---

El Departamento de Prueba necesitaba una aplicación para la gestión del Proceso de Evaluación de Software de los Centros de Producción en la UCI. Una vez efectuada la investigación correspondiente al proceso de evaluación en el Departamento de Prueba y de las nuevas herramientas, metodologías y tecnologías existentes en el mundo, se obtuvo una herramienta automatizada en la que se incrementan los servicios y la documentación necesaria para los usuarios del Departamento. Garantizando un mejor aseguramiento de la calidad de los productos de software en la UCI. Se usó una metodología relativamente nueva Groovy and Grails, que facilitaba el trabajo con la Web.

La herramienta facilita la navegación y posibilita a los roles autorizados del sitio editar todo el contenido necesario. La aplicación da la posibilidad de mostrar toda la información relacionada con el proceso de evaluación de una forma sencilla e intuitiva de acuerdo a los objetivos trazados.

Esta herramienta fue ponencia en el Evento de Ciencia y Técnica, mostrando gran interés hacia su culminación y despliegue por lo útil y práctico de su utilización. Se espera que cumpla finalmente con las expectativas creadas

Los objetivos propuestos en el presente proyecto fueron cumplidos de forma satisfactoria de acuerdo con el cronograma presentado, mostrando avances paulatinos de acuerdo al mismo, formando conocimientos sólidos, basados en bibliografías actualizadas y de actores con prestigio. Concluyendo un objetivo general materializado y una serie de recomendaciones, para proyectos futuros.

## **RECOMENDACIONES**

---

En el presente trabajo se recomienda:

- ❖ Realizar un estudio más profundo sobre el lenguaje de desarrollo de aplicaciones Groovy and Grails, ya que es una herramienta novedosa y de probada factibilidad, que está tomando auge para la construcción de aplicaciones Web.
- ❖ Ampliar las funcionalidades de la herramienta, que posibilite un mejor servicio por parte del Departamento de Pruebas.
- ❖ Convertir este Sistema en un producto fuertemente negociable con el objetivo de vender el servicio a empresas interesadas
- ❖ Perfeccionar el diseño de la herramienta para futuras versiones.
- ❖ Ampliar la gama de modelos de Calidad con los que ha de trabajar la parte Evaluadora.
- ❖ Se recomienda que este trabajo sea tomado como material de estudio para aquellas personas que vayan a realizar una aplicación similar o un trabajo referente al tema expuesto.

## BIBLIOGRAFÍA

---

**Beginning, Judd y Christopher, M. 2009.** *Groovy and Grails*. New York : s.n., 2009.

*Descripción y características de este potente y moderno.* **Álvarez, Angel Miguel. 2001.** 2, Estados Unidos de América : s.n., 2001, Vol. I.

**Dickinson, Jon. 2009.** *Grails 1.1 Web Application Development*. missoula : Packt Publishing Ltd, 2009.

*Diseño del Sistema de tarjetas de Crédito con UML.* **Navarro, J M. 2003.** s.l. : Universidad Nacional Mayor de San Marcos, 2003.

**Gil, Elizabeth. 2005.** *Lenguaje JAVA*. 2005.

**Graeme, Rocher y Brown, Jeff. 2009.** *The Definitive Guide to Grails Second Edition*. United States of America : s.n., 2009.

**REYNOSO, C. 2005.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Argentina : Buenos Aires, 2005.

**Santi Paz, Pedro Luis. 2008.** *Perl I*. 2008.

**SCHEMULLER, J. 2000.** *Aprendiendo UML en 24 horas*. México : MedinaPromen 542, 2000.

*Sistema ANACONDA para el análisis automático de la calidad del software.* **Martín, Enrique y Sáenz, Fernando. 2009.** Madrid, España : s.n., 2009.

**Soria, A.** El maravilloso mundo de los portales. *Cap 3 Portales especializados*. s.l. : AcademicPress.

**U.S. Commerce Department. 2003.** Software and Systems Division. *Software and Systems Division*. [En línea] 7 de Noviembre de 2003. [Citado el: 22 de Noviembre de 2009.] <http://www.itl.nist.gov/div897>.

**Wesley, Addison. 2005.** *El Proceso Unificado de Desarrollo*. s.l. : June 248, 2005.



## BIBLIOGRAFÍA REFERENCIADA

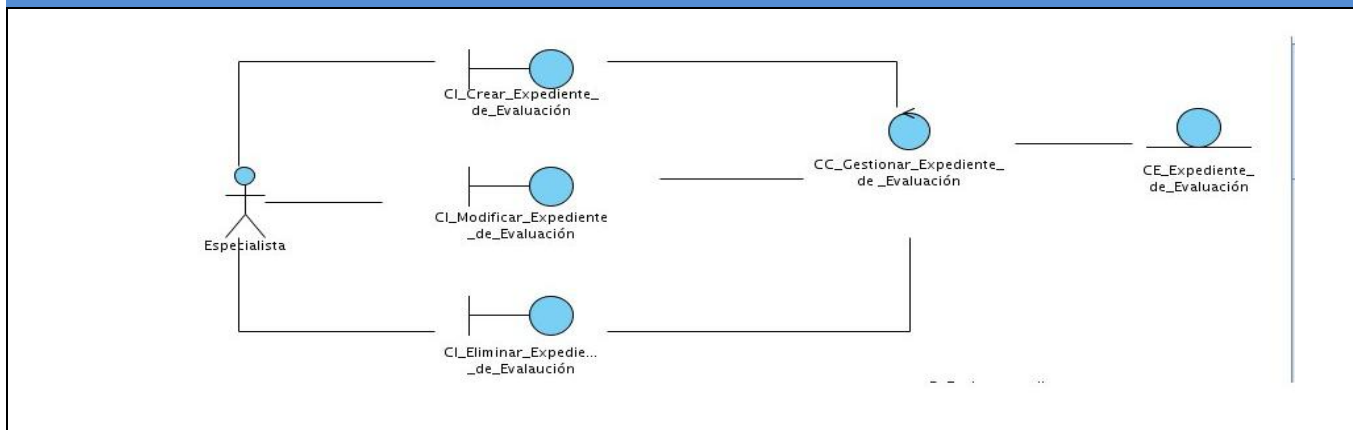
---

- Bashar, Abdul Jawad. 2010.** Groovy and Grails Recipes Grails-NetBeans Integration. 2010.
- Booch, G, Rumbaugh, J y Jacobson, I. 1999.** *El Lenguaje Unificado de Modelado*. s.l. : Addison-Wesley, 1999. ISBN: 8478290362 ISBN-13: 9788478290369.
- Casanovas, J. 2005.** *Usabilidad y arquitectura del software*. s.l. : Ra-Ma , 2005. ISBN: 8478976523 ISBN-13: 9788478976522.
- Christopher, Judd M, Faisal Nusairat, Joseph y Shingler, James. 2008.** Beginning Groovy and Grails from Novice to Professional. 2008.
- Corsi, M. 2010.** Beneficios de las Aplicaciones Web. *Aplicaciones Web*. s.l. : Springer 596, 2010.
- Joaquín, José. 2010.** *Calidad y Medición de Sistemas de Información*. s.l. : VSP. ternational Science Publishers, 2010.
- Larman, C. 1999.** UML y Patrones. *Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999.
- MANJARRES V, A. (04/2009).** Tipos de Portales. *Desarrollando Web Primera edición*. s.l. : Anaya Multimedia, (04/2009).
- N, Miguel. 2007.** Portales de Informática. *Características de los Portales*. s.l. : Freeman, W.H. 235, 2007.
- Pressman, R. 1999.** *Software Engineering. A Practitioner's Approach*. USA : McGraw – Hill, 1999.
- Saavedra López, Esteban. 2010.** Grails: Framework para el desarrollo de Aplicaciones Web. *6ta Parte*. 2010.
- . 2010. Grails: Framework para el desarrollo de Aplicaciones Web. *1ra Parte*. 2010.
- Smith, Glen y Ledbrook, Peter. 2010.** Grails in Action. 2010.

**ANEXO 1: DIAGRAMAS DE CLASES DE ANÁLISIS**

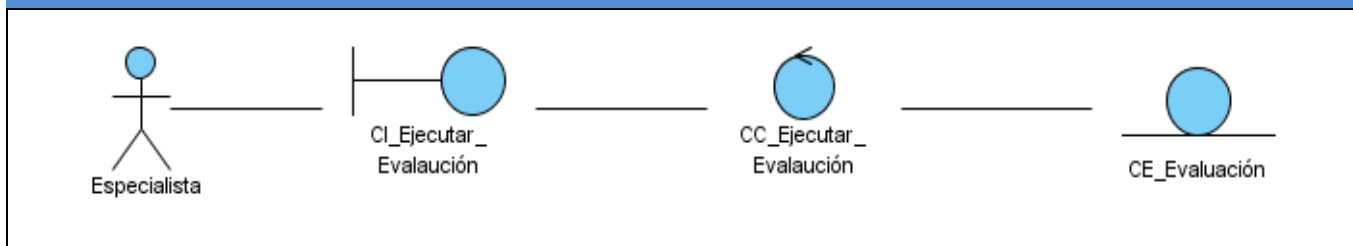
Se recomienda para consultar el resto de los diagramas de clases del análisis, visitar la versión ampliada de la presente tesis.

**Diagrama de clases de Análisis (CU\_ Gestionar Expediente de Evaluación)**



**Figura 8.** Diagrama de clases de Análisis (CU\_ Gestionar Expediente de Evaluación)

**Diagrama de clases de Análisis (CU\_ Ejecutar Evaluación)**



**Figura 9.** Diagrama de clases de Análisis (CU\_ Ejecutar Evaluación)

**ANEXO 2: DIAGRAMAS DE SECUENCIA DEL DISEÑO**

Se recomienda para consultar el resto de los diagramas de secuencia del diseño, visitar la versión ampliada de la presente tesis.

## Diagrama de Secuencia (Interacción) Esc\_Crear Expediente de Evaluación

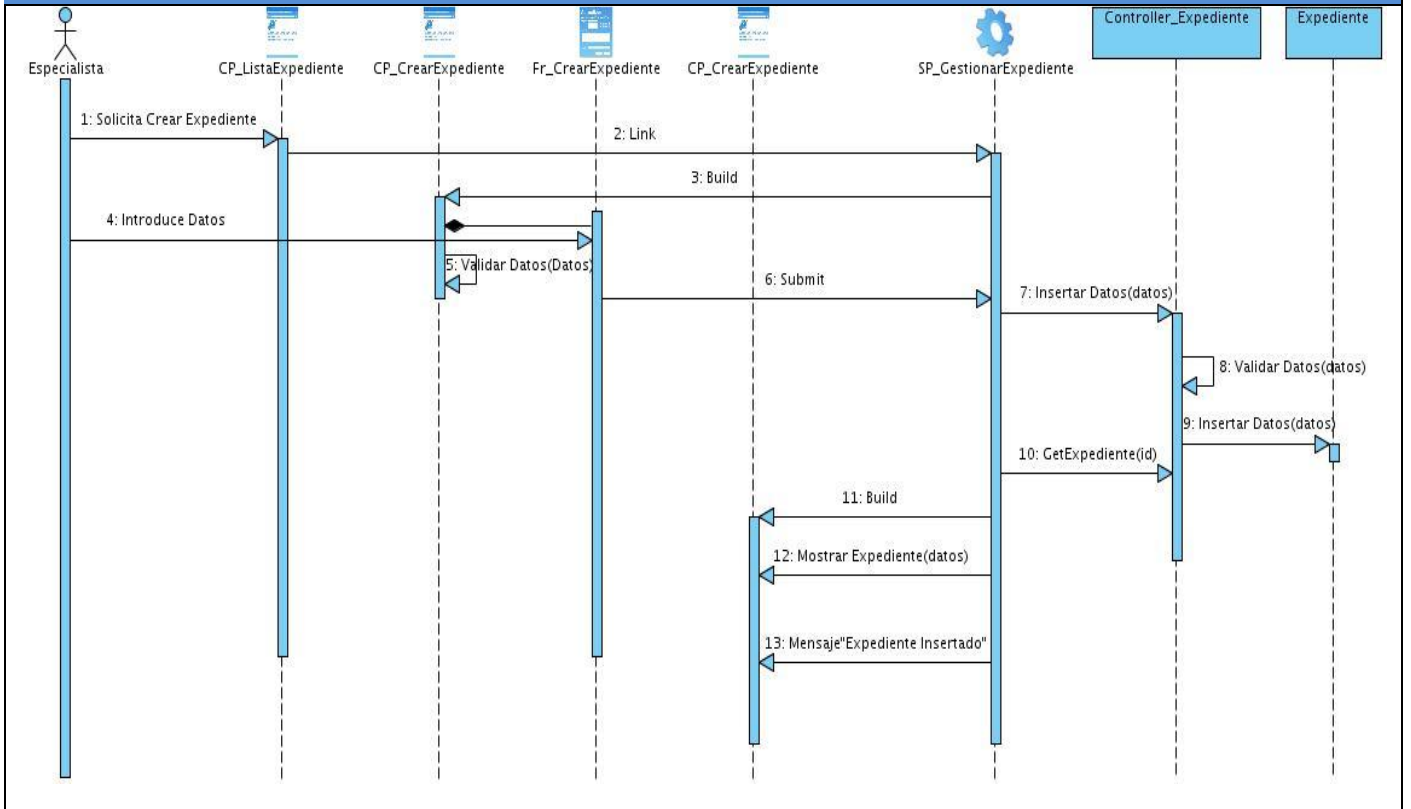


Figura 10. Diagrama de Secuencia (Interacción) Esc\_Crear Expediente de Evaluación

## Diagrama de Secuencia (Interacción) Esc\_Modificar Expediente

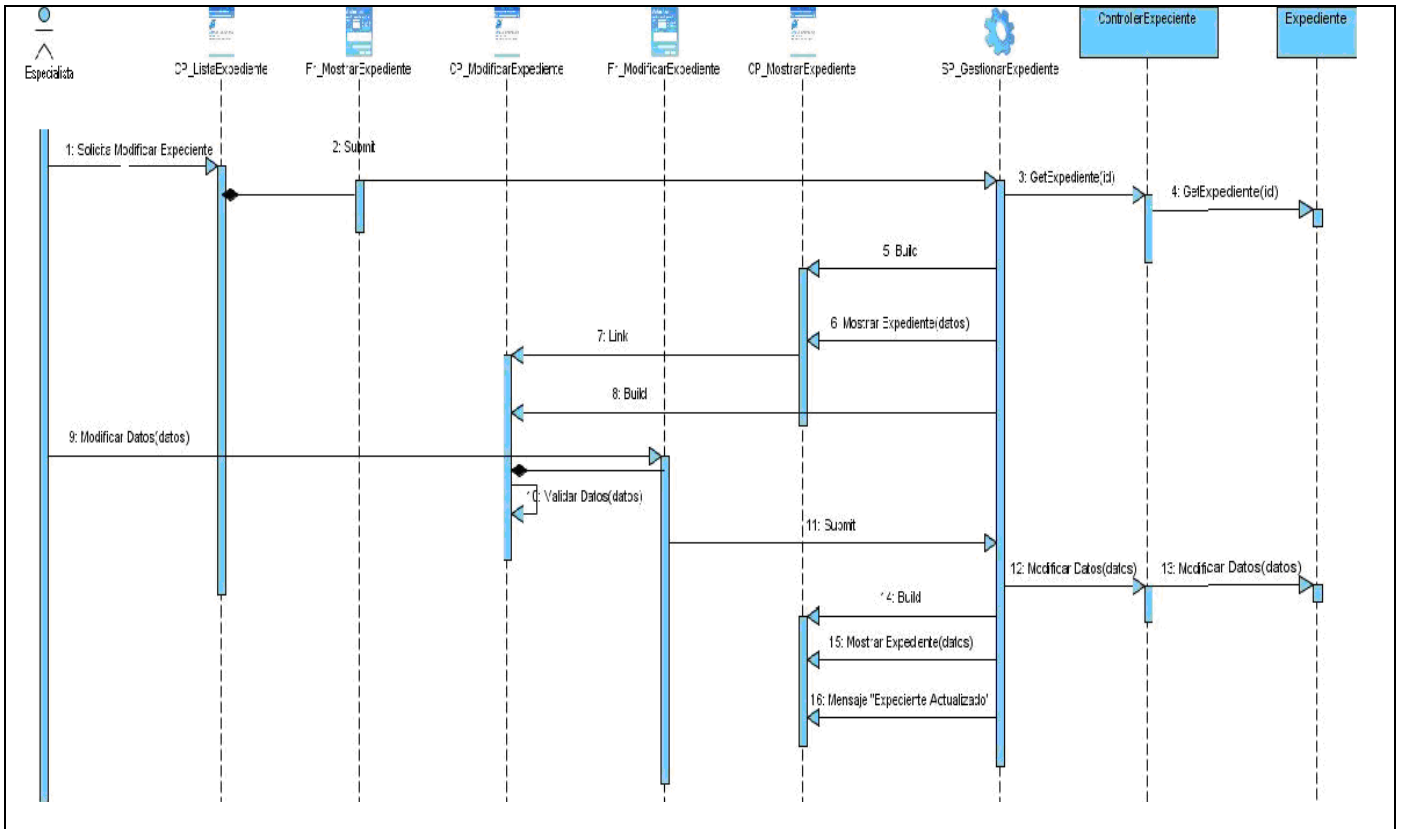


Figura 11. Diagrama de Secuencia (Interacción) Esc\_Modificar Expediente

**Diagrama de Secuencia (Interacción) Esc\_Eliminar Expediente**

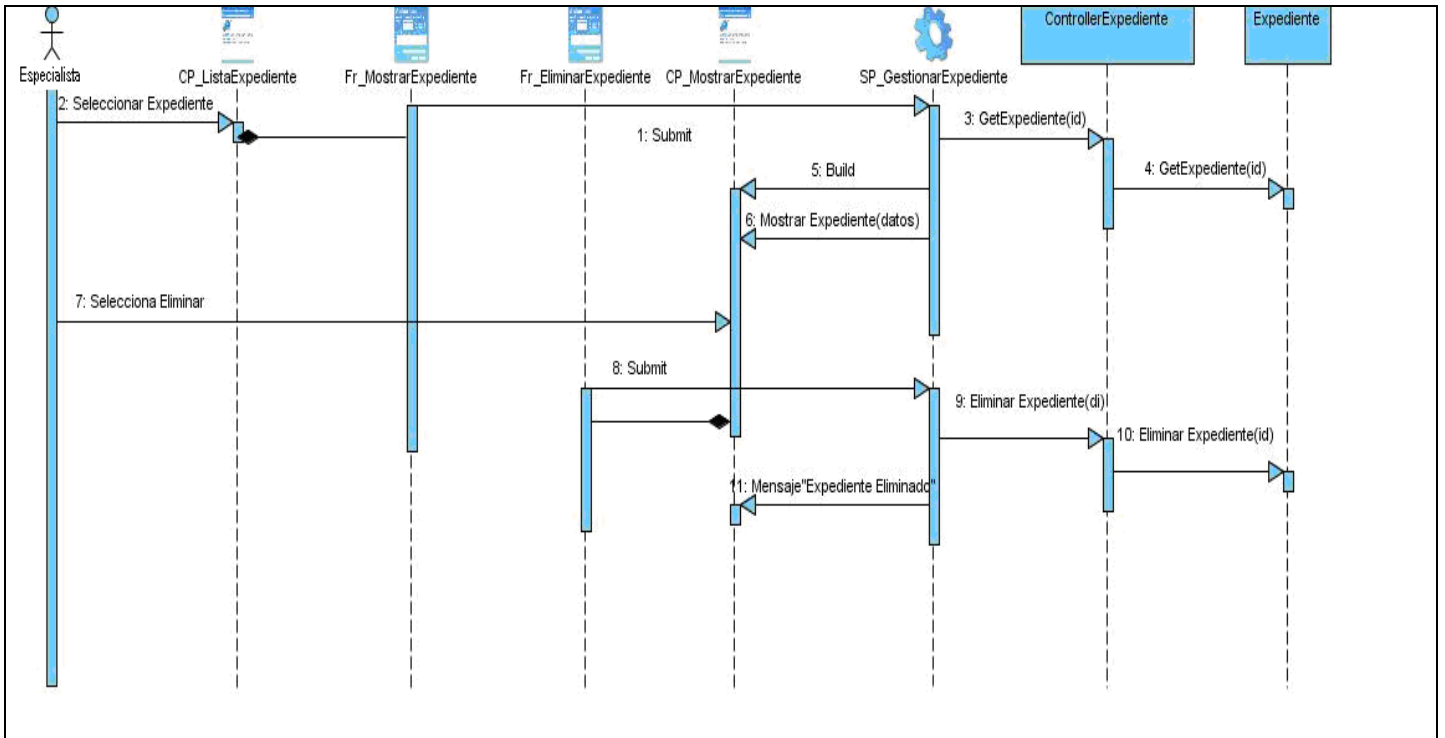
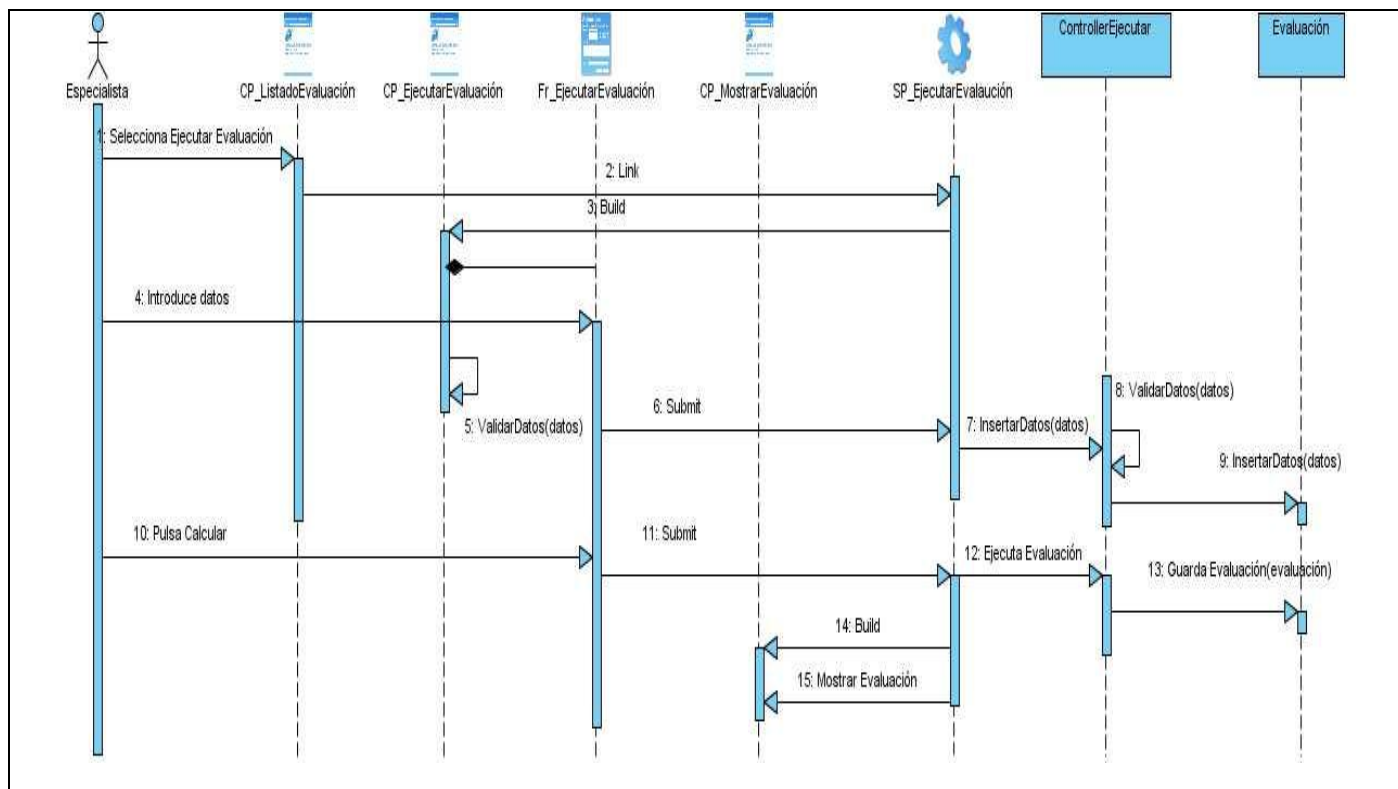


Figura 12. Diagrama de Secuencia (Interacción) Esc\_Eliminar Expediente

**Diagrama de Secuencia (Interacción) Esc\_Ejecutar Evaluación**



**Figura 13.** Diagrama de Secuencia (Interacción) Esc\_ Ejecutar Evaluación

**ANEXO 3: DIAGRAMA DE CLASES DE DISEÑO WEB**

Se recomienda para consultar el resto de los diagramas de clases de diseño Web, visitar la versión ampliada de la presente tesis.

**Diagrama de clases del Diseño(CU\_ Gestionar Expediente de Evaluación)**

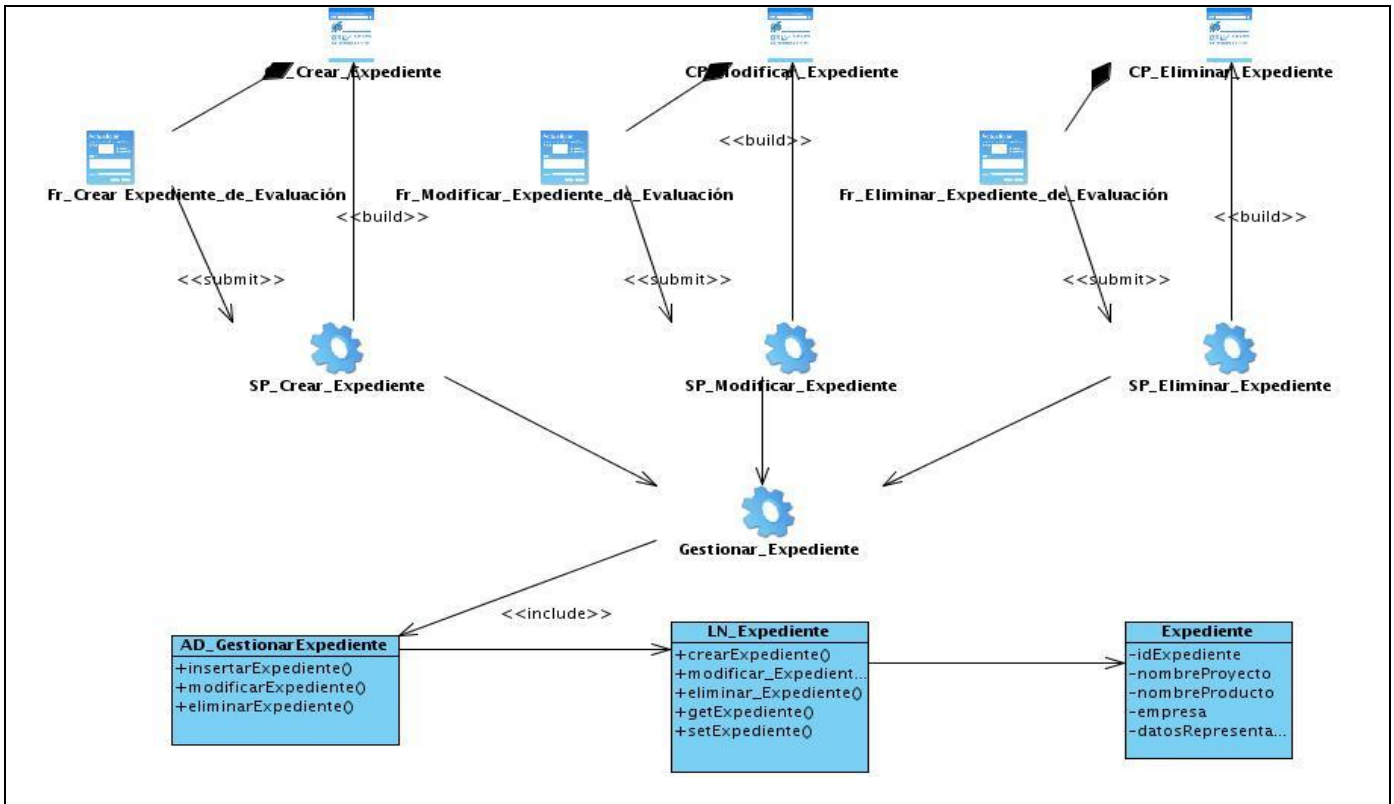


Figura 14. Diagrama de clases del Diseño (CU\_Gestionar Expediente de Evaluación)

**Diagrama de clases del Diseño(CU\_Ejecutar Evaluación)**

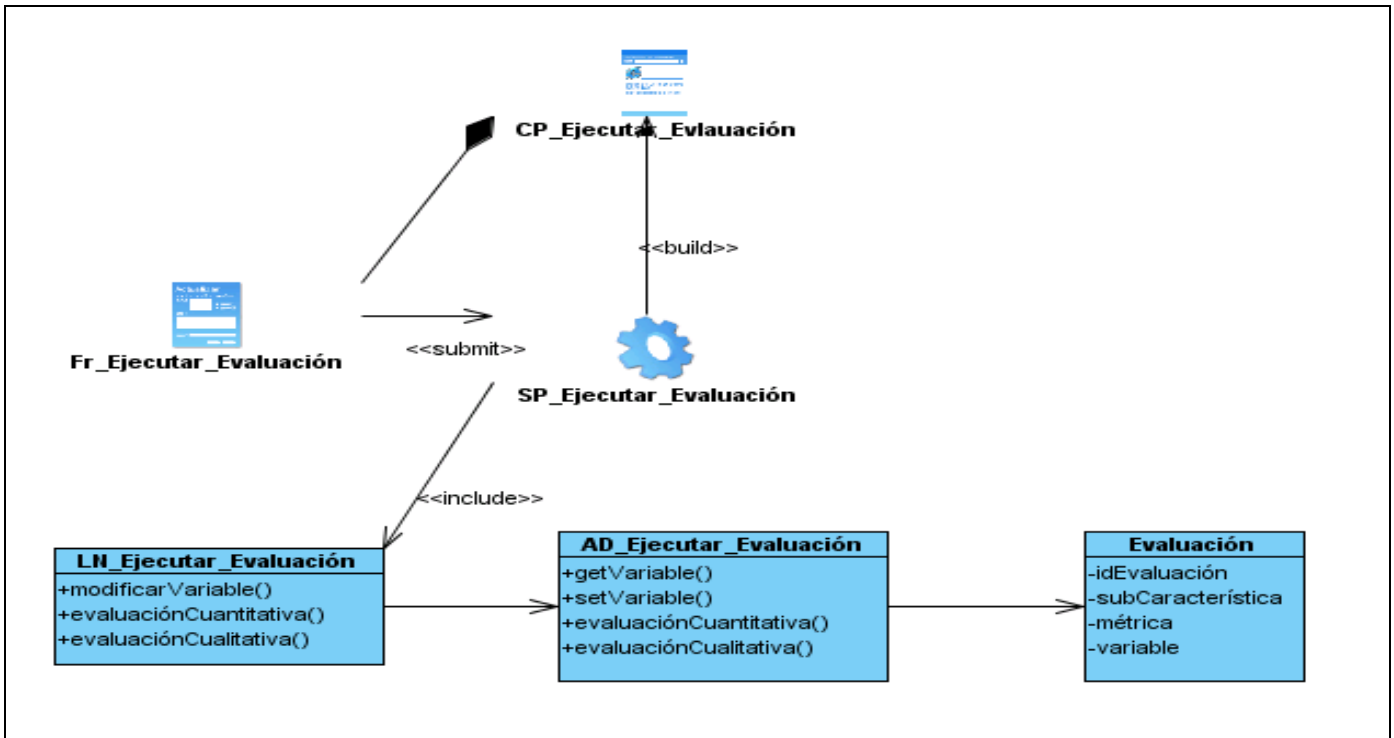


Figura 15. Diagrama de clases del Diseño (CU\_Ejecutar Evaluación)