

**Universidad de las Ciencias Informáticas**  
**Facultad 5**



**Título: Implementación de un portal empresarial para  
el Centro de Consultoría y Desarrollo de  
Arquitecturas Empresariales.**

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autor:** Armando Masó Mosqueda

**Tutor:** Ing. Orestes Febles

**Co-tutor:** Ing. Maité Rivero Cruz

Junio, 2012

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 \_\_\_\_\_ del año \_\_\_\_\_.

Armando Masó Mosqueda

Orestes Febles Díaz

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

DATOS DE CONTACTO

**Tutor:**

Ing. Orestes Febles Díaz  
Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba  
Correo: ofebles@uci.cu

**Co-Tutor:**

Ing. Maite Rivero Cruz  
Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba  
Correo: mriveroc@uci.cu

## AGRADECIMIENTOS

A mis padres por ser quienes me impulsan a ser mejor y a confiar en mí. Agradezco a mi novia por soportarme durante este año tan intenso de trabajo, a mis amigos por ayudarme a cumplir con este sueño.

A mi prima por siempre salir en mi defensa y por el apoyo tecnológico.

A Maité por hacer prácticamente suya mi tesis brindándome su ayuda en todo momento.

A Orestes por su papel tan importante como tutor.

A Yadira por ser prácticamente mi otra cotutora.

A los profesores del Centro por ayudarme desinteresadamente cuando necesite de ellos.

A Franklin por todas las molestias que le ocasione durante el desarrollo del trabajo.

Y a todos los que me alentaron a llegar hasta aquí.

¡Muchas Gracias!

## DEDICATORIA

A todos aquellos que como mi mamá, mi papá, mis hermanos, mi novia y amigos han estado siempre presentes en mi vida apoyándome y se han ganado mi cariño y admiración.

## RESUMEN

En la Universidad de las Ciencias Informáticas existen un número considerable de portales que contribuyen a mantener actualizada a toda la comunidad universitaria, además de posibilitar el intercambio de conocimientos. En el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales perteneciente a la facultad 5 se necesita la incorporación de un sistema que permita gestionar de una manera sencilla la información, conocimientos y aplicaciones referentes al centro en las diferentes esferas, y mantenga informados y actualizados a sus miembros. Se realizó la investigación que lleva como título “Implementación de un portal empresarial para el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales”, con el objetivo de desarrollar un portal que permita centralizar la información, el conocimiento y las aplicaciones desarrolladas.

Para el logro del objetivo propuesto se realizó una amplia revisión bibliográfica, que permitió elaborar y exponer la fundamentación teórica de los portales web, así las herramientas y tecnologías utilizadas. El proceso de desarrollo del portal fue guiado por la metodología de desarrollo XP, utilizando Java como lenguaje de programación. La construcción del sistema propuesto tiene un carácter iterativo, permitiendo la inserción de nuevas funcionalidades y la corrección de errores.

**PALABRAS CLAVE:** gestión, información, portal empresarial, portal web.

## TABLA DE CONTENIDOS

DECLARACIÓN DE AUTORÍA .....	I
DATOS DE CONTACTO.....	II
AGRADECIMIENTOS .....	III
DEDICATORIA .....	IV
RESUMEN.....	V
TABLA DE CONTENIDOS.....	VI
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
<b>1.1 Introducción</b> .....	4
<b>1.2 Web 2.0</b> .....	4
<b>1.3 Portales Web</b> .....	4
1.3.1 Clasificaciones de los Portales .....	5
1.3.2 Portales Corporativos o Empresariales.....	5
1.3.3 Uso de las Aplicaciones Enriquecidas en Internet (RIA) en los portales. ....	7
1.3.4 Uso de Portales en Cuba y en la UCI. ....	7
<b>1.4 Selección de las tecnologías y herramientas que se utilizarán en el desarrollo de la solución.</b> 8	
1.4.1 Servidor de Aplicaciones .....	8
1.4.2 Plataformas corporativas para la construcción de portales. ....	8
1.4.3 Sistema Gestor de Bases de Datos (SGBD) .....	13
1.4.4 Herramientas Case .....	14
1.4.5 Herramienta de pruebas.....	15
<b>1.5 Portlets</b> .....	16
<b>1.6 Aplicaciones Compuestas</b> .....	16
<b>1.7 Uso en la Universidad de las Ciencias Informáticas</b> .....	16
<b>1.8 Metodología</b> .....	17
1.8.1 ¿Qué es una metodología de desarrollo?.....	17
1.8.2 Metodologías tradicionales.....	17
1.8.3 Metodologías ágiles .....	17
1.8.4 Selección de la metodología.....	18

1.8.4 Roles. Metodología XP .....	21
1.8.5 Fases de la Metodología Extreme Programming (XP) .....	23
<b>1.9 Arquitectura de la Información</b> .....	<b>26</b>
<b>1.10 Conclusiones del capítulo</b> .....	<b>26</b>
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA</b> .....	<b>27</b>
<b>Introducción</b> .....	<b>27</b>
<b>2.1 Arquitectura del portal (Fig. 1)</b> .....	<b>27</b>
2.1.1 Presentación .....	28
2.1.2 Lógica del negocio .....	28
2.1.3 Datos.....	28
2.1.4 Patrón Arquitectónico utilizado .....	28
2.1.5 Descripción de los componentes utilizados. ....	29
<b>2.2 Arquitectura de la Información</b> .....	<b>30</b>
2.2.1 Definición de la audiencia.....	30
2.2.2 Clasificación de la audiencia .....	30
2.2.3 Necesidades de la audiencia .....	31
2.2.4 Mapa de navegación .....	32
2.2.5 Estructura.....	32
2.2.6 Elementos del sistema de navegación .....	33
2.2.7 Diseño de la estructura de las pantallas tipo .....	34
<b>2.3 Características del portal</b> .....	<b>34</b>
<b>2.4 Seguridad en el portal</b> .....	<b>36</b>
<b>2.5 Metodología Extreme Programming (XP)</b> .....	<b>36</b>
2.5.1 Fase 1: Exploración.....	36
2.5.2 Fase 2: Planificación de la Entrega .....	38
2.5.3 Fase 3: Iteraciones .....	39
2.5.4 Fase 4: Producción .....	45
<b>2.6. Conclusiones del capítulo.</b> .....	<b>51</b>
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA</b> .....	<b>52</b>
<b>3.1 Introducción</b> .....	<b>52</b>



---

<b>3.2 Implementación</b> .....	52
3.2.1 Iteración 1 .....	52
<b>3.3 Pruebas</b> .....	59
3.3.1 Pruebas de aceptación.....	60
3.3.2 Pruebas de rendimiento. ....	63
<b>3.4 Conclusiones del capítulo</b> .....	66
CONCLUSIONES .....	67
RECOMENDACIONES .....	68
BIBLIOGRAFÍA.....	69
ANEXOS.....	71
Anexo 1. Encuesta .....	71
Anexo 2. Historias de Usuarios. ....	73
Anexo 3. Tareas de la Iteración 1.....	79
Anexo 4. Tareas de la Iteración 2.....	80
Anexo 5. Tareas de la Iteración 3.....	89
Anexo 6. Pruebas de aceptación del sistema.....	100
Anexo 7. Pruebas de rendimiento del sistema.....	112
GLOSARIO .....	125

## INTRODUCCIÓN

El aumento de la productividad del personal es un tema que siempre ha llamado el interés de las empresas, con el fin de optimizar el tiempo y mejorar la calidad de sus productos. La integración de sistemas, la creación de espacios compartidos de trabajo para la colaboración, los servicios de mensajería y correo electrónico; son herramientas que se utilizan con este fin.

Muchas empresas, instituciones y organismos, han visto en los portales una potente herramienta con la que se puede gestionar información, conocimientos, aumentar la productividad y reducir al mismo tiempo los costos de implementación. Son de mucha utilidad si lo que desea es integrar sistemas, gestionar contenidos y ganar público a fines con los intereses de una determinada institución, proporcionándole una variedad de servicios que los haga preferir el producto que se le ofrece, para lo cual la organización se debe apoyar en la inmensa retroalimentación que ofrecen elementos como los foros, las wikis y las comunidades.

Los Portales Empresariales o Corporativos son una clasificación que se les otorga a los portales atendiendo a la evolución que han presentado y a las características que han adquirido a través de esta evolución. Se especializan en temas de diferentes índoles, entre los más comunes se pueden encontrar salud, deportes, finanzas, música, entre otros.

La herramienta principal que manejan las organizaciones para relacionarse con los usuarios de la red es el website o sitio web corporativo. Esto posibilita que la organización pueda comunicarse con su público tanto interno como externo, para ello puede utilizar las mismas técnicas comunicativas que en los medios de tradicionales. Varias empresas han pasado de tener un sitio web a un portal corporativo, esto constituye un elemento que ayuda a optimizar las relaciones de la compañía, a través de la organización lógica de las páginas web y enlaces o links que contienen. Entre las distintas ventajas que ofrece un Portal Corporativo en relación con un sitio web se encuentra la reducción de costos asociados a la información (creación, impresión, distribución...), una comunicación más personalizada, rápida e interactiva, mayor claridad y transparencia organizativa, transmisión del conocimiento, un entorno de comunicación estratégico, creación de valor para empleados, socios y clientes”[1].

A pesar de tener muy pocos años de experiencia, el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), ya cuenta con importantes logros a nivel nacional e internacional, lo cual beneficia de manera sustancial a la universidad y al país, este centro, presenta la siguiente **situación problemática**: carece de una herramienta que sea capaz de gestionar la información, conocimientos y las aplicaciones relacionadas con el CDAE. No cuenta con un espacio para la colaboración entre sus trabajadores y estudiantes que le permita intercambiar ideas o dudas relacionadas con el trabajo que se desempeña. Esto trae como consecuencia que la información necesaria para trabajar se encuentra dispersa y complejiza su acceso.

Dada la situación problemática antes expuesta, se plantea como **problema de la investigación** ¿Cómo lograr una plataforma que permita la gestión de la información, el conocimiento y el despliegue de aplicaciones en el CDAE?

El desarrollo de este trabajo presenta como **objeto de estudio** las plataformas de portales empresariales, y teniendo como **campo de acción**: el proceso de gestión de plataformas de portales empresariales y desarrollo de aplicaciones basadas en la plataforma.

El **objetivo de la investigación** es elaborar una propuesta que favorezca la gestión de la información, el conocimiento y aplicaciones relacionadas con el centro.

**La investigación se plantea como objetivos específicos.**

- Estudiar las tendencias de los portales empresariales.
- Desarrollar una propuesta de un portal empresarial.
- Validar la solución desarrollada.

La investigación tiene como **idea a defender**:

La propuesta desarrollada permitirá favorecer la gestión de la información, los conocimientos y las aplicaciones desarrolladas en el centro.

Para lograr un óptimo desarrollo y cumplimiento de los objetivos planteados se hará uso de un conjunto de métodos científicos, los cuales se presentan a continuación:

Como método teórico se utilizó el análisis **histórico lógico**; empleándose en la investigación y análisis de los portales prestando especial atención a los de tipo Empresarial.

Se utilizaron como métodos empíricos:

La **entrevista** se utiliza en varios encuentros que se planificaron con el cliente para conocer diferentes aspectos relacionados con las funcionalidades presentes en el portal, así como para la elaboración del su diseño gráfico.

La **encuesta** se utilizó para investigar aspectos importantes acerca del uso de la herramienta seleccionada en la universidad.

El contenido de este trabajo de tesis estará estructurado en tres capítulos, en los cuales se abordarán las peculiaridades del proceso de investigación y desarrollo del Portal Empresarial.

En el Capítulo 1: se explica qué son los Portales Empresariales, sus ventajas y desventajas. También se hace alusión a su uso tanto a nivel internacional, nacional como en la universidad. Se hace además mención de la metodología de desarrollo y las tecnologías seleccionadas para darle solución al problema planteado anteriormente.

En el Capítulo 2: Se realiza una propuesta de arquitectura para el sistema, también se elabora una descripción de las necesidades del sistema mediante historias de usuarios. Además se muestra, el modelo de datos, para una mejor comprensión de la información que será manipulada.

En el Capítulo 3: Se lleva a cabo la descripción de las tareas de implementación generadas por cada historia de usuario realizadas en las tres iteraciones propuestas, obteniendo en cada una de ellas una versión del producto, así como las pruebas de aceptación y de rendimiento efectuadas sobre el portal.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### **1.1 Introducción.**

En este capítulo se aborda el tema de los Portales Empresariales, sus ventajas y desventajas, su uso tanto a nivel internacional, nacional y universitario. Se hace además mención de la metodología de desarrollo y las tecnologías seleccionadas para darle solución al problema planteado anteriormente. Además se explicarán conceptos relacionados con el objeto de estudio para un mejor entendimiento de los siguientes capítulos.

### **1.2 Web 2.0.**

En la web las tecnologías presentan un avance acelerado con relación a otros tipos de tecnología en la rama de la informática, son renovadas o mejoradas en breves espacios de tiempo. La web en sus inicios era un entorno estático que hacía uso de páginas HTML e imágenes GIF para lograr cierta animación en sus contenidos y presentaba muy poca actualización de sus datos e interacción con los usuarios, esta es la llamada Web 1.0. La Web 2.0 es la representación de la evolución tomando como punto de partida la Web 1.0, donde las aplicaciones web son enfocadas al usuario final, dándole a este el control de las mismas; haciendo uso de AJAX, DHMTL, XML, SOAP, entre otras que han ido surgiendo con el transcurrir de los años.

### **1.3 Portales Web.**

Muchas de estas mejoras concernientes a la Web 2.0 se ven reflejadas en los Portales Web; estos son sitios web que ofrecen de forma integrada, una amplia variedad de servicios y recursos al usuario. Por lo general, los portales ofrecen noticias actualizadas al instante, buscadores, foros para compartir opiniones, chat, juegos online, tiendas virtuales para realizar compras electrónicas y servicio de correo electrónico, teniendo como intención básica que el usuario pueda satisfacer todas sus necesidades en un mismo sitio [2]. Ofreciendo una amplia gama de opciones, de forma tal que puedan convertirse en la “puerta de ingreso” del internauta a la Web. Se supone que el usuario tendrá al portal en su página de inicio y utilizará todos sus servicios antes de comenzar a navegar en otros sitios más específicos. Yahoo, Terra, Google y MSN son algunos ejemplos de portales de Internet aunque algunos de estos como Google están especializados en búsquedas.

Entre las características que poseen los portales, se pueden mencionar la implicación directa de los usuarios en la creación de comunidades y la generación de contenidos siendo este mucho más interactivo, pasando de ser unidireccional a bidireccional; características que hace posible una mejor retroalimentación y de la cual se debe hacer uso para definir el camino a seguir, siendo el usuario quien orienta y guía según sus necesidades, lo cual posibilita un ambiente más colaborativo.

### **1.3.1 Clasificaciones de los Portales.**

Con el transcurso de la evolución de los portales han ido surgiendo características con relación a sus contenidos, la cuales se pueden agrupar en dos grandes bloques [3]:

- Genéricos, también denominados Horizontales (Horizontal Enterprise Portal, HEP), e incluso megaportales. Intentan mostrar de manera estructurada todo lo que podría encontrarse en la Red. Aun así, algunos permiten cierto grado de personalización.
- Especializados o Verticales (Vertical Enterprise Portal, VEP). Suelen centrarse exhaustivamente en un aspecto dentro de un ámbito geográfico o de una temática (finanzas, actualidad, salud, etc.), o bien adaptan sus contenidos a las necesidades del cliente (tanto en información como en servicios). En este último caso el acceso puede no ser gratuito y se trata de una intranet. Los Portales Corporativos deben considerarse de forma que dentro de una intranet de empresa proveen a sus miembros de información interna y externa relacionada con la propia entidad.

### **1.3.2 Portales Corporativos o Empresariales.**

Son una extensión de las intranets empresariales, donde se proveerá de información y materiales de trabajo relacionados con la empresa a empleados e incentivarán el acceso por parte de clientes, socios e interesados que compartan los mismos intereses que la empresa, siempre cuidando la información que se está brindando en función de quien la esté requiriendo.

Un Portal Empresarial es "...una herramienta desarrollada con tecnología web que se asemeja al sistema nervioso del cuerpo humano y que permite llevar la información importante a los usuarios, a través de un punto de acceso único, de tal forma que toda la información de la empresa se accede desde una dirección

*url* única, reduciendo la incertidumbre de dónde buscarla. Dependiendo del usuario, el portal corporativo le permite visualizar la información que corresponde con sus funciones dentro de la empresa, lo que le ayuda a enfocar sus acciones y mejorar su productividad, al mismo tiempo que asegura un mejor aprovechamiento de la misma para la toma de decisiones” [4]. Definiendo para este concepto un grupo de características, ventajas y desventajas:

**Características:**

- Proporciona un punto de acceso único para toda la información.
- Cuenta con un mecanismo de autenticación único para el usuario.
- Integra la información de todos los sistemas de la empresa.
- Cuenta con mecanismos para realizar búsquedas de información.
- Adapta la ubicuidad y el acceso a la información.
- Incorpora procesos colaborativos que permiten al usuario identificar prioridades.
- Proporciona herramientas para la colaboración de equipos a distancia.
- Integra la información externa de proveedores, entorno, clientes, etc., mediante mecanismos de contenido sindicado.

**Ventajas:**

- Optimiza los recursos informáticos de la empresa.
- Reduce tareas redundantes.
- Mejora la comunicación y el enfoque de los usuarios en sus tareas cotidianas.
- Mejora la seguridad de la información
- Garantiza la interconectividad en la información.

**Desventajas:**

- Necesidad de reestructurar los sistemas existentes.
- Cambio de paradigma para toda la organización.
- La reducción de tareas implica la reestructuración de la organización.

### **1.3.3 Uso de las Aplicaciones Enriquecidas en Internet (RIA) en los portales.**

Las tecnologías llamadas RIA son la nueva generación de aplicaciones web cuyo principal foco de atención es el usuario, es un nuevo tipo de aplicación con más ventajas que las tradicionales aplicaciones Web, entre ellas se pueden destacar la mejora de su arquitectura para lograr efectos eficientes y acortar la brecha entre las aplicaciones Web y las de escritorio.

En los entornos RIA no se producen recargas de página, ya que desde el principio se carga toda la página y los datos necesarios para que la aplicación funcione, y se puede discutir si se cargan o no datos que en un futuro no muy lejano estaremos necesitando. Sólo se produce comunicación con el servidor cuando los datos son requeridos, cuando se necesitan datos externos como datos de una base de datos o de otros ficheros externos [5].

Es una realidad que esta nueva generación de aplicaciones web trae grandes ventajas tanto para el uso de desarrolladores como para los usuarios. Este tipo de tecnología se puede explotar para lograr así un mejor rendimiento del portal a desarrollar, mejorando de esta forma su interactividad con el servidor ya que el uso de una tecnología RIA mejora el rendimiento con los servidores, también los usuarios se adaptan mejor a los cambios, estos se le hacen más intuitivos y efectivos.

### **1.3.4 Uso de Portales en Cuba y en la UCI.**

En Cuba, muchas empresas e instituciones hacen uso de este medio, algunos para difundir la verdad del país en todo el mundo; los portales como Cuba Debate, Juventud Rebelde, Vanguardia, Portal Cuba, son claros ejemplos de esto; retomando las palabras de Valdés N. al contrario de lo que se asegura o asume frecuentemente, la mayoría de “las páginas web tratan de tópicos comerciales, económicos y culturales, y las páginas dedicadas estrictamente a temas políticos son pocas”[6].

Portal Cuba es un excelente portal donde se puede dirigir y encontrar cualquier información relacionada con el país, desde política, cultura, deportes, turismo, por solo citar algunos.

En UCI es muy común el uso de portales, cada una de las facultades, sedes y también algunas comunidades de desarrollo hacen uso para llevar las noticias que allí se generan a la comunidad de estudiantes y profesores, siendo nuestra intranet uno de los portales por excelencia para la comunidad.



También la universidad cuenta con un portal en Internet encargado de difundir nuestra los logros alcanzados en el duro batallar para mantenernos siendo una “Universidad de excelencia”. Es importante señalar que estos portales centran sus servicios en la gestión de contenidos, no son sistemas capaces de gestionar aplicaciones.

#### **1.4 Selección de las tecnologías y herramientas que se utilizarán en el desarrollo de la solución.**

##### **1.4.1 Servidor de Aplicaciones.**

Un servidor de aplicaciones consiste en un contenedor que abarca la lógica de negocio de un sistema, y que provee respuestas a las peticiones de distintos dispositivos que tienen acceso a ella. Son un claro ejemplo del modelo cliente-servidor, cuyo lado cliente ejecuta requerimientos de procesamiento al otro lado, donde el servidor se encarga de procesar y responder[7].

Existen muchos tipos de servidores de aplicación, basados en distintas tecnologías. Los más comunes funcionan en arquitecturas J2EE, aunque también los hay en otras variantes. Ejemplos de estos son:

- Oracle application server.
- IBM WebSphere.
- JBoss.
- Apache Tomcat.

##### **1.4.2 Plataformas corporativas para la construcción de portales.**

Cuba se encuentra enmarcada en una dura batalla por la migración hacia el software libre y por lo tanto UCI como puntera de vanguardia en esta esfera, y siguiendo esta línea de trabajo si se va a desarrollar un portal se deben buscar las herramientas de código abierto (Open Source) necesarias para acometer esta tarea con la calidad requerida.

Diversas son las herramientas para la construcción de portales, dentro de las que se encuentran algunas propietarias como:

- Microsoft SharePoint Portal Server
- Blakord Portal
- Oracle9iAS Portal
- Websphere Portal

Herramientas de gran calidad pertenecientes a empresas como Microsoft, Oracle e IBM, pero por las cuales se debe pagar un precio por su uso, además del costo por soporte técnico; en cambio existen otras diseñadas con el mismo fin pero en este caso Open Source y de una calidad comparable con estos productos antes mencionados, se estudiarán con el fin de dar una solución óptima y de una excelente calidad al problema presentado por el CDAE.

- jBoss Portal.
- Liferay Portal.

### **jBoss Portal.**

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en java puro. Al estar basado en java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible java e implementa todo el paquete de servicios de J2EE y considerado por Gartner como líder en su cuadrante mágico.

#### **Características:**

- Producto Open Source.
- Escalable.
- Presenta un alto desempeño.
- Presenta una Arquitectura Modular.
- Producto de licencia de código abierto sin coste adicional (LGPL).
- Cumple los estándares.
- Confiable a nivel de empresa.

### **Liferay Portal.**

Creado en 2000 Liferay Portal es una plataforma web corporativa que posibilita desarrollar soluciones empresariales apoyándose en un conjunto de estándares entre los cuales se encuentran la adopción de estándares en las tecnologías empleadas para desarrollo de aplicaciones llamadas portlets, es una herramienta multiplataforma, de igual forma permite la gestión de contenidos y documentos, la edición web y espacios de trabajo compartidos, la colaboración a nivel de empresa, redes sociales, la creación de

portales corporativos entre otra posibilidades que lo hacen el portal de vanguardia de Open Source. Utiliza en su versión estándar la licencia de software libre empleada por el Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology) permitiendo utilizar y modificar la aplicación libremente sin costo alguno. Su versión empresarial presenta una licencia del mismo tipo [8].

Generalmente sólo utilizan el término de gestor de contenidos para referirse a este Portal Server, pero la realidad es muy distinta, a pesar de que uno de los tantos usos que brinda Liferay es la gestión de contenidos, es capaz también de gestionar aplicaciones, presenta compatibilidad con estándares como Java Specification Request por sus siglas en inglés (JSR-268) y Servicios Web para Portlets Remotos (en inglés Web Services for Remote Portlets WSRP) y una disponibilidad de 60 portlets pre-construidos.

En su instalación básica contiene numerosos portlets o funcionalidades (aplicaciones), es independiente de la plataforma en que se desee trabajar –“sistemas operativos o bases de datos”. Liferay cuenta un una gran comunidad de usuarios, por solo citar algunos datos en 2011 creció cerca de un 40 por ciento, hasta alcanzar los 56.000 miembros con una media de 75.000 descargas mensuales y un total de instalaciones que superan los 3,2 millones de descargas en todo el mundo, lo cual beneficia enormemente el trabajo con esta herramienta ya que posibles errores y problemas que surjan se les puede dar solución inmediatamente, gracias a la activa participación de la comunidad.

### **Especificaciones técnicas de Liferay.**

Liferay presenta un inmenso conjunto de especificaciones técnicas, que lo convierten en una opción por excelencia para desarrollar soluciones Empresariales. Gracias a su estrecho compromiso con los estándares presenta tecnologías muy potentes, aportándole un increíble nivel de solides; entre las más utilizadas se puede mencionar: Velocity, Spring 3.0 & AOP, Hibernate, Java J2EE/JEE, AJAX; soportando lenguajes de scripting como JavaScript, Ruby, PHP y Python. Basa su rendimiento y escalabilidad al uso de clustering, balanceo de cargas e integración con otros sistemas.

La seguridad es un tema que sin dudas no puede faltar a la hora de desarrollar un portal y Liferay utiliza tecnologías de cifrado de última generación y basadas en estándares, incluyendo algoritmos avanzados como DES, MD5 y RSA. Ha sido probado y situado entre las plataformas de portal más seguras empleando la suite Logiscan de LogicLibrary, para la gestión de identidades hace uso de la autenticación

y la sincronización con servidores de Protocolo de acceso a Directorios Ligeros (LDAP, del inglés Lightweight Directory Access Protocol). Su arquitectura presenta un sistema jerárquico y extensible de comunidades y organizaciones siendo estas replicables mediante plantillas.

### Cuadro comparativo entre Liferay Portal y jBoss Portal.

<b>Criterio</b>	<b>Liferay</b>	<b>jBoss</b>
Aplicaciones por defecto.	Alrededor de 60	Muy pocas
Comunidades y organizaciones.	Soporta	No soporta
Páginas personales por usuarios.	Soporta	No soporta
Lenguajes.	Soporta 22 lenguajes	No soporta
Wikis	Soporta	No soporta
Blogs	Soporta	No soporta
Política de contraseñas	Soporta	No soporta
Mensajería instantánea	Soporta	No soporta
Autorización basada en roles	Soporta	Soporta
Autenticación SSO	Soporta	Soporta

**Tabla 1. Diferencias entre Liferay Portal y jBoss Portal [9].**

Otras de las grandes diferencias de estas dos plataformas radica en:

- JBoss Portal sólo se puede instalar en el servidor de aplicaciones JBoss mientras que Liferay funciona en cualquier contenedor J2EE o el contenedor de servlets.
- Las características de CMS son muy mínimos en JBoss Portal, todo lo que se puede hacer es crear páginas HTML con un editor de texto enriquecido y hacer algunas actividades de manipulación de ficheros (crear el directorio, mover y copiar archivos y así sucesivamente).
- La gestión de contenidos en Liferay es más rica en características que en JBoss, se pueden crear tipos de contenido personalizados y plantillas propias. Esto ayuda a separar el contenido de su presentación. También es muy fácil de crear nuevas páginas y añadir automáticamente a la barra de navegación.

- Liferay cuenta con una la comunidad de usuarios más grande y además presenta muchos más clientes empresariales que jBoss.
- A pesar de Liferay trabaja con casi cualquier servidor de aplicaciones, su instalación tardan más tiempo considerando que la instalación de JBoss Portal es un proceso simple.
- Liferay cuenta con un IDE de desarrollo, que está respaldado por un Kit de desarrollo de software (SDK).

La plataforma Liferay Portal es la seleccionada para desarrollar el portal del CDAE, dado que presenta un conjunto de características y las facilidades para la implementación de aplicaciones que posibilitan un rápido y óptimo desarrollo del sistema. La selección del lenguaje de programación, el entorno de desarrollo integrado, el framework RIA para el desarrollo de componentes visuales y el framework para el Mapeo Objeto Relacional (ORM), fue en consecuencia a la decisión tomada.

### **Lenguaje de Programación. Java**

Es un lenguaje compilado de programación orientado a objetos, las clases que genera son interpretadas por una máquina virtual, siendo esta la que mantiene el control sobre las clases que se estén ejecutando. Es multiplataforma, independiente del sistema operativo en el que se trabaje; cuando la máquina virtual de java está ejecutando algún código, realiza comprobaciones de seguridad, lo cual permite que los errores sean tratados en el momento que son percibidos.

### **Entorno de desarrollo integrado. Eclipse**

Es un entorno de desarrollo integrado capaz de proveer un conjunto de herramientas para administrar espacios de trabajo, construir, correr y depurar aplicaciones, además posee una arquitectura de módulos llamados plug-ins que le permite incorporarle múltiples funcionalidades, como lo es la instalación de un plug-ins para la creación de portlets. Es un entorno de desarrollo diseñado para ser extendido con sofisticadas herramientas[10]. Soporta el plug-ins Liferay IDE, este plug-ins permite el desarrollo de temas, portlets, hooks y plug-ins para los portales creados con Liferay.

### **Framework RIA para el desarrollo de componentes visuales**

Vaadin Plug-ins

Es un framework de componentes para desarrollar aplicaciones web construidas sobre Google Web Toolkit (GWT). Los componentes son similares a los widgets de GWT pero su estado se almacena en el servidor. La sincronización entre el servidor y los componentes es manejada automáticamente por el framework[11]. Facilita la creación de portlets y hooks para Liferay sin romper con los estándares cumplidos por el servidor de aplicaciones.

### **Framework Mapeo Objeto Relacional (ORM)**

Hibernate

Cuando se trabaja bajo el paradigma “Orientado a objetos” y a la vez se utiliza el de sistemas de bases de datos relacionales, los desarrolladores afrontan las desavenencias de estos dos enfoques de trabajo. Un ORM es un framework que propone una nueva forma de modelar los datos, permitiendo solucionar la diferencia que existe entre estos dos paradigmas [12]. Hibernate es una herramienta ORM de código abierto muy madura y completa. Tiene como objetivo facilitar el mapeo de atributos entre una base de datos relacional y un modelo de objetos, a través de archivos (XML) o con anotaciones en las entidades, además de ser una de las tecnologías integradas a Liferay.

### **1.4.3 Sistema Gestor de Bases de Datos (SGBD).**

PostgreSQL

Es un sistema gestor de base de datos objeto-relacional, bajo licencia BSD (en inglés Berkeley Software Distribution). Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. Es el sistema de gestión de bases de datos de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre [13].

Características de PostgreSQL:

Atomicidad (Indivisible): es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.

Consistencia: es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.

Aislamiento: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.

Durabilidad: es la propiedad que asegura que una vez realizada la operación, esta persistirá y no se podrá deshacer aunque falle el sistema [13].

### MySQL.

Es un sistema de gestión de bases de datos relacional, que utiliza el lenguaje de programación Structured Query Language (SQL) y entre sus características se destacan:

Es multiplataforma. Tiene APIs disponibles para C, C++, Java, Perl, PHP, Python, Ruby, entre otras más. Proporciona sistemas de almacenamientos transaccionales y no transaccionales. Presenta un sistema de privilegios y contraseñas que es muy flexible y seguro, y permite verificación basada en el host. Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix [14]. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo, presenta además una gran portabilidad entre sistemas y soporta diferentes tipos de datos en para las columnas así como hasta 32 índices por tablas.

Al finalizar la comparación de estos dos SGBD se llegó a la conclusión de utilizar PostgreSQL debido a su condición de ser multiplataforma, por estar licenciado bajo GNU y tener gran número de características, como lo son: soportar una capacidad de almacenamiento en el orden de los TB (terabytes), poseer gran escalabilidad y buen rendimiento puede soportar una mayor cantidad de peticiones concurrentes, así como también la de almacenar procedimientos en la propia base de datos, también permite la crear subconsultas, características que no se encuentran presente en MySQL.

#### **1.4.4 Herramientas Case. Visual Paradigm .**

Es una herramienta multiplataforma que utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del software (análisis y diseño orientados a objetos, construcción, pruebas y despliegue). Permite

realizar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas, entre muchas otras funcionalidades. Presenta una excelente documentación la herramienta UML CASE basada en abundantes tutoriales de UML, demostraciones interactivas y proyectos UML.

Visual Paradigm ofrece un diseño enfocado al negocio, permitiendo así generar un software de mayor calidad. Presenta una licencia Open Source a diferencia de su homólogo Rational Rose.

#### **1.4.5 Herramienta de pruebas.**

##### JMeter

Es una herramienta java de uso gratuito, que permite realizar pruebas de rendimiento y funcionales sobre aplicaciones web. Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Permite además activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario [15].

Algunas de las funcionalidades que brinda la herramienta se encuentran:

- Permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y WebServices (en Beta).
- Permite realizar pruebas distribuidas en distintos ordenadores que actuarán como clientes simulando varios hilos que harán función de usuarios.
- Permite generar un caso de prueba es a través de una navegación de usuario.

##### LoadRunner

Es una herramienta para realizar pruebas de carga de Mercury Interactive que permite pre-visualizar el comportamiento y el rendimiento del sistema. Además, permite poner a prueba toda la infraestructura corporativa para identificar y aislar los posibles problemas mediante la simulación de la actividad de miles de usuarios. Realiza pruebas en toda la infraestructura corporativa, que comprende las soluciones e-business y las aplicaciones personalizadas, simulando la actividad de miles de usuarios, con lo que los equipos de desarrollo de aplicaciones y sitios Web pueden mejorar el rendimiento de las aplicaciones.

Es la herramienta de pruebas de carga más escalable que permite simular la actividad de miles de usuarios con los mínimos recursos de hardware. Se integra a la perfección con las herramientas de



gestión del rendimiento de Mercury Interactive. Los mismos scripts creados durante las pruebas pueden volverse a utilizar para monitorizar la aplicación una vez terminada su implantación. Presenta una interfaz API abierta, con la que los usuarios y otros fabricantes pueden integrar LoadRunner en sus propios entornos[16]. Puede ser empleado en sistemas operativos Windows y Unix, siendo su licencia propietaria.

JMeter es la herramienta seleccionada debido a las características mencionadas que presenta, lo que permiten hacer un análisis exhaustivo de las pruebas realizadas. Además de ser la más completa y útil de las herramientas gratis.

### **1.5 Portlets.**

Los portlets son pequeñas aplicaciones web escritas en Java que se ejecutan en una parte de una página web, forman el corazón del portal, ya que son los que contienen todas sus funcionalidades. Estas mini aplicaciones pueden ser locales o remotas al portal, una de sus principales características es que son totalmente autónomos. Estos pueden ser desarrollados utilizando cualquier framework de java que soporte el desarrollo de portlets, también bajo las especificaciones de Liferay MVCPortlets y AlloyPortltes [17].

### **1.6 Aplicaciones Compuestas.**

Son aplicaciones construidas reutilizando la lógica de dos o más aplicaciones existentes para formar una nueva aplicación sin tener que empezar de cero. Una aplicación compuesta consiste de funcionalidad tomada de diferentes fuentes aprovechando el concepto de la Arquitectura Orientada a Servicios (SOA). Los componentes pueden ser servicios web individuales o una selección de funcionalidad de otras aplicaciones o aplicaciones completas existentes de las cuales se toman sus resultados empaquetados como servicios web[18].

### **1.7 Uso en la Universidad de las Ciencias Informáticas.**

Los resultados que a continuación se muestran fueron el resultado de una encuesta aplicada en la UCI, que tuvo como objetivo obtener información con respecto al nivel de conocimiento y familiarización de Liferay Portal y sus tecnologías asociadas. [Ver Anexo 1.](#)

En la UCI se han impartido cursos para promover el desarrollo con esta herramienta, pero su uso no ha proliferado mucho, el conocimiento de este Portal Server es poco, aunque se ha desarrollado con él en la universidad.

## **1.8 Metodología.**

### **1.8.1 ¿Qué es una metodología de desarrollo?**

Una metodología de desarrollo posibilita definir un conjunto de reglas a seguir por parte de los desarrolladores de software para tener un proyecto con la más alta calidad posible, este conjunto de reglas contienen los procedimientos que se realizarán en cada proceso del proyecto, así como las técnicas y herramientas a utilizar y los documentos en los que se auxiliarán para seguir este proceso en todo el ciclo de vida del software.

### **1.8.2 Metodologías tradicionales.**

Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. Entre las metodologías tradicionales o pesadas podemos citar a RUP (por sus siglas en inglés Rational Unified Process).

### **1.8.3 Metodologías ágiles.**

Las metodologías ágiles se crean como una mejora de las metodologías tradicionales ya que tienen una serie de características o ventajas frente a las metodologías tradicionales, mejoran en gran medida la forma en que se desarrollan los proyectos utilizando metodologías tradicionales debido a que estas necesitan un gran esfuerzo por parte del equipo de desarrollo para realizar proyectos poco productivos y son poco flexibles a los cambios de requisitos mientras que las ágiles incluyen el desarrollo iterativo que permite cambios en los requisitos y con la comunicación el equipo puede tomar decisiones a lo largo del

ciclo de vida del proyecto y aplicarlas inmediatamente. La siguiente tabla hace referencia a otros beneficios de las metodologías ágiles con respecto a las tradicionales.

### Comparación entre metodologías ágiles y tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

**Tabla 2. Diferencias entre metodologías ágiles y tradicionales [17].**

#### 1.8.4 Selección de la metodología.

##### Metodología RUP.

Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo. Su propósito es asegurar la producción de software de alta calidad que se

ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles. RUP divide el proceso de desarrollo en 4 fases, definiendo para cada una de estas fases una o varias iteraciones siguiendo un modelo de cascada para los flujos de trabajo que requieran las fases[19].

### **Fases.**

- Inicio.
- Elaboración.
- Construcción.
- Transición.

### **Flujos de trabajo.**

- Modelado del negocio.
- Análisis de requisitos.
- Análisis y diseño.
- Implementación.
- Pruebas.
- Despliegue.
- Gestión de configuración y Cambios.
- Gestión de Proyecto.
- Ambiente.

### **Las características principales de RUP**

- Guiado por casos de uso.
- Centrado en arquitectura.
- Iterativo e Incremental.
- Desarrollo basado en components.
- Utilización de un único lenguaje de modelado.
- Proceso Integrado.
- Define un conjunto de roles, actividades y artefactos.

Es un proceso enorme y general, por lo que antes de usarlo debe ser adaptado a las características de la empresa, se basa en casos de uso para describir lo que se espera del software y está orientado a la arquitectura del sistema, documentándose lo mejor posible.

### Metodología XP

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega es inminente. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto[19].

### Características de XP

La metodología se basa en:

- Pruebas unitarias y de aceptación.
- Refactorización
- Programación en pares.

### ¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

### Comparación entre las metodologías XP y RUP.

	XP	RUP
<b>Tamaño de los equipos.</b>	Proyectos y equipos pequeños.	Proyectos y equipos grandes.
<b>Obtención de los requisitos.</b>	Historias de usuarios.	Casos de usos.

<b>Carga de trabajo.</b>	Proceso ligero.	Proceso pesado.
<b>Relación con el cliente.</b>	Comunicación fluida con el cliente.	Se presentan los artefactos al final de una fase.
<b>Tipo de desarrollo</b>	Iterativo e incremental.	Iterativo e incremental.
<b>Código fuente</b>	Compartición de código.	Propiedad de código.

**Tabla 3. Diferencias entre las metodologías XP y RUP.**

La metodología (XP) comparada con otras metodologías ágiles es una de las mejores según una serie de aspectos tratados. XP es la mejor documentada, con una amplia comunidad y certificada, además de ser la más utilizada por las empresas en proyectos de software.

Comparándola con una metodología tradicional como lo es RUP caracterizada por tener un desarrollo interno en etapas iterativas, centrado en la arquitectura, dirigido por casos de uso y se utiliza para realizar grandes y complejos proyectos, XP resulta ser ideal para el desarrollo de este proyecto ya que se utiliza principalmente en proyectos y equipos pequeños, tiene un desarrollo iterativo e incremental, interactúa con el usuario final y los requisitos pueden ser cambiados, mientras que RUP interactúa con el usuario estratégico y solo se permiten cambios en una sola fase. De acuerdo a ventajas RUP está más documentada pero XP permite la simplicidad de código, una mejor comunicación y una alta calidad en el producto en un mínimo período de tiempo.

Por las razones antes expuestas la metodología XP es la seleccionada para guiar el proceso de desarrollo del portal.

#### **1.8.4 Roles. Metodología XP.**

Existen diferentes variaciones de roles para esta metodología en numerosas fuentes, por lo que nos apoyaremos en la propuesta original de Kent Beck para definir el conjunto de roles.

##### **Programador.**

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo[20].

**Cliente.**

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema[20].

**Encargado de pruebas (*Tester*).**

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas[20].

**Encargado de seguimiento (*Tracker*).**

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.

También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración[20].

**Entrenador (*Coach*).**

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente[20].

**Consultor.**

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico[20].

**Gestor (Big boss).**

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación[20].

**1.8.5 Fases de la Metodología Extreme Programming (XP).****Fase 1: Exploración.**

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología[20].

**Historia de Usuario.**

La Historia de Usuario se emplea para cada una de las características que va a tener el sistema, son las acciones que se deben implementar en el sistema. Cada una de estas es implementada en una determinada iteración del proyecto, de las cuales también hacemos referencia a los puntos estimados que son los que nos definen el tiempo de duración de las Historias de Usuario a implementar.

Después de hacer un estudio de las Historias de Usuario que son de interés para la primera etapa de entrega del producto se definió la siguiente estructura para obtener los datos de las Historias de Usuarios determinadas.

Historia de Usuario	
Nombre:	
Número:	Usuario:
Dependiente:	
Prioridad:	Puntos Estimados:
Descripción:	

Tabla 4. Estructura de una Historia de Usuario.



**Nombre:** El nombre de la Historia de Usuario (HU).

**Número:** Identificador de la HU.

**Dependiente:** Si depende de la implementación de otra HU, si depende se le debe especificar el identificador de dicha HU si no depende de ninguna se le pone "No".

**Prioridad:** Es la importancia que tiene la HU en el sistema, se clasifican en cuanto a Baja, Media y Alta.

**Usuario:** El que determina la HU.

**Puntos Estimados:** puntos estimados que son los que nos definen el tiempo de duración de las Historias de Usuario a implementar donde 1 punto estimado equivale a 4 días de trabajo.

### **Fase 2: Planificación de la Entrega.**

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días[20].

### **Fase 3: Iteraciones.**

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción[20].

### **Fase 4: Producción.**

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y

las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento) [21].

Para la realización de esta fase se hace una relación entre las clases, su responsabilidad y el colaborador ya sean el tipo de clases o las clases relacionadas. Estos datos serán mostrados mediante la siguiente tabla:

Nombre de la clase	
Responsabilidades	Colaboradores

**Tabla 5. Relación entre las clases.**

**Clase:** es cualquier evento que realizara una serie de acciones que se le especificara para obtener, modificar, eliminar, adicionar datos y otras según los resultados que se quieran obtener.

**Responsabilidades:** de una clase es la lógica del negocio asociada y las que realizan sus atributos y métodos.

**Colaboradores:** son el resto de las clases con las que trabaja para llevar a cabo sus responsabilidades.

#### **Fase V: Mantenimiento.**

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura[21].

#### **Fase VI: Muerte del Proyecto.**

El cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la

documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo[21].

### **1.9 Arquitectura de la Información.**

El término "Arquitectura de la Información" (AI) fue utilizado por primera vez por Richard Saul Wurman en 1975, quién la define como:

Es el estudio de la organización de la información con el objetivo de permitir al usuario encontrar su vía de navegación hacia el conocimiento y la comprensión de la información[22].

Al enfocar la Arquitectura de la Información solamente a la web, esta es la disciplina de estructurar, organizar, clasificar la información que se publica en sitios, portales e intranet, con el simple pero importante propósito de que el usuario encuentre y utilice satisfactoriamente la información que necesita.

Para una correcta selección de la Arquitectura de la Información que se empleará en el portal se tuvieron presentes una serie de aspectos como lo son la audiencia y el contenido a publicar.

### **1.10 Conclusiones del capítulo.**

En el capítulo se expusieron características y conceptos importantes que se deben tener presentes durante todo el ciclo de vida del producto, así como las herramientas y tecnologías que serán utilizadas, seleccionando la Metodología XP para el desarrollo, empleando Liferay Portal como plataforma corporativa para el desarrollo del portal, utilizando java como lenguaje de programación y como entorno de desarrollo eclipse por ser el plug-ins Liferay IDE soportado por este y por las características que presenta. Como sistema gestor de base de datos PostgreSQL es el seleccionado; siendo Visual Paradigm elegida para el modelado y jMeter para realizar las pruebas de carga y estrés al portal. Estas herramientas brindan la posibilidad de crear un producto robusto y competitivo, además se realizó una breve descripción del estado del arte acerca del uso de Liferay en el marco nacional y universitario.

## CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

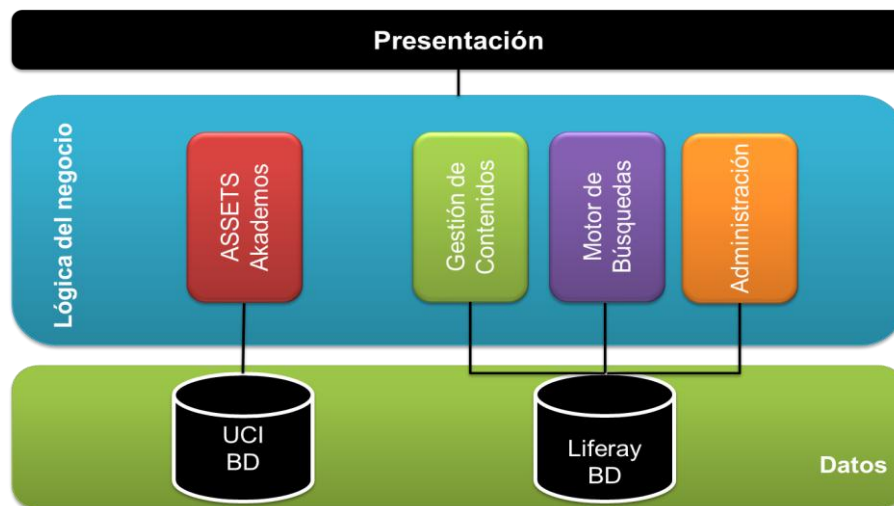
**Introducción.**

Durante el desarrollo de este capítulo se define la arquitectura que se empleará en el desarrollo del portal, se describen las necesidades del sistema mediante historias de usuarios, además se muestran, el modelo de datos para una mejor comprensión de la información que será manipulada y se aborda algunos aspectos de la arquitectura de la información que se especifica para el portal.

**2.1 Arquitectura del portal (Fig. 1).**

La Arquitectura Orientada a Servicios es la arquitectura por referencia de Liferay, convirtiéndose así en una plataforma para el desarrollo y la integración estructurada. Liferay Portal posee una arquitectura basada en un sistema jerárquico y extensible de comunidades y organizaciones, las cuales son replicables mediante plantillas. Las comunidades son un entorno de trabajo colaborativos, en el que todos los usuarios comparten conocimientos de un tema común, siendo la actividad de estos en la comunidad la que le da mantenimiento a la misma.

La arquitectura del portal estará concebida en términos de presentación, lógica del negocio y el acceso a datos, estando estrechamente vinculadas estas tres capas.



**Fig. 2. Arquitectura propuesta para el portal CDAE.**

Los servicios y contenidos que se presentan en el portal se desarrollan como arquitecturas verticales sobre estas capas. Es el caso de los sistemas de información, el motor de búsqueda y los portlets de administración así como los desarrollados como valor añadido, debido a que estos requieren de lógica y datos propios.

### **2.1.1 Presentación.**

En la capa de presentación se establece la composición de los contenidos que se presentan al usuario, es la responsable de todos los aspectos relacionados con la interfaz de usuario como resultado de su petición o al ingresar al sistema. La navegabilidad, el formateo de la información que le es mostrada como resultado de alguna acción al internauta y la validación de datos, son cuestiones que se tratan en esta capa.

### **2.1.2 Lógica del negocio.**

Es la parte del portal encargada de atender las peticiones de los usuarios. Es aquí donde se establecen todas las reglas del negocio que deben cumplirse, de ahí que ubicamos todas las clases controladoras del sistema garantizando una mejor estructura con respecto a la lógica del negocio. La capa de presentación se comunica con esta capa, para acceder a las solicitudes y presentar los resultados, y a su vez, la capa de negocio accede a la capa de acceso a datos, para solicitar la información necesitada por el usuario o almacenarla, en dependencia de la solicitud antes hecha.

### **2.1.3 Datos.**

Es donde se almacena de forma persistente toda la información necesaria para facilitar satisfacer las peticiones de los usuarios mediante los servicios ofrecidos por el portal. El perfil de los usuarios del portal, contenidos de agregación como los blogs y wikis e informaciones de las publicaciones.

### **2.1.4 Patrón Arquitectónico utilizado.**

**Patrón de diseño DAO:** Plantea como solución utilizar un Data Access Object (DAO) para abstraer y encapsular todos los accesos a la fuente de datos. Creando una capa separada que se ocupa del acceso a datos con total independencia del gestor y la base de datos, lo que brinda la posibilidad de trabajar con

varios gestores y bases de datos dentro de la misma aplicación. El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos.

### **2.1.5 Descripción de los componentes utilizados.**

**Cliente:** Los usuarios pueden acceder al portal desde dispositivos tradicionales (pc) e inalámbricos (pc, móviles). Los desarrolladores pueden acceder al portal de la API expuesta a través de un conjunto de clases de túneles ofrecidas por el despliegue del Tunnel-Web incluido en el portal.

**Servidor de Aplicaciones (Tomcat):** El servidor de aplicaciones tendrá alojado el Portal el cual consumirá información de servidores universitarios, como es el caso del servidor LDAP y los servicios de ASSETS y Akademos publicados en la UDDI de la universidad.

**Lightweight Directory Access Protocol (LDAP):** Proporciona un repositorio único y centralizado para todas las aplicaciones (single logon) posibilitando el acceso a la información de usuarios, grupos, perfiles, directorio de servicios y entidades. Ofrece el acceso más eficiente a los datos del directorio en comparación con estructuras basadas en Sistemas de Gestión de Bases de Datos Relacionales (RDBMS). Garantía de transportabilidad futura de la información almacenada en LDAP al tratarse de un estándar con amplia implantación.

**Servicios de Akademos y ASSETS:** Ofrecen información detallada de los profesores, estudiantes y departamentos con los que cuenta el Centro, posibilitando de este modo el fácil acceso a esta a través de los Servicios Web publicados.

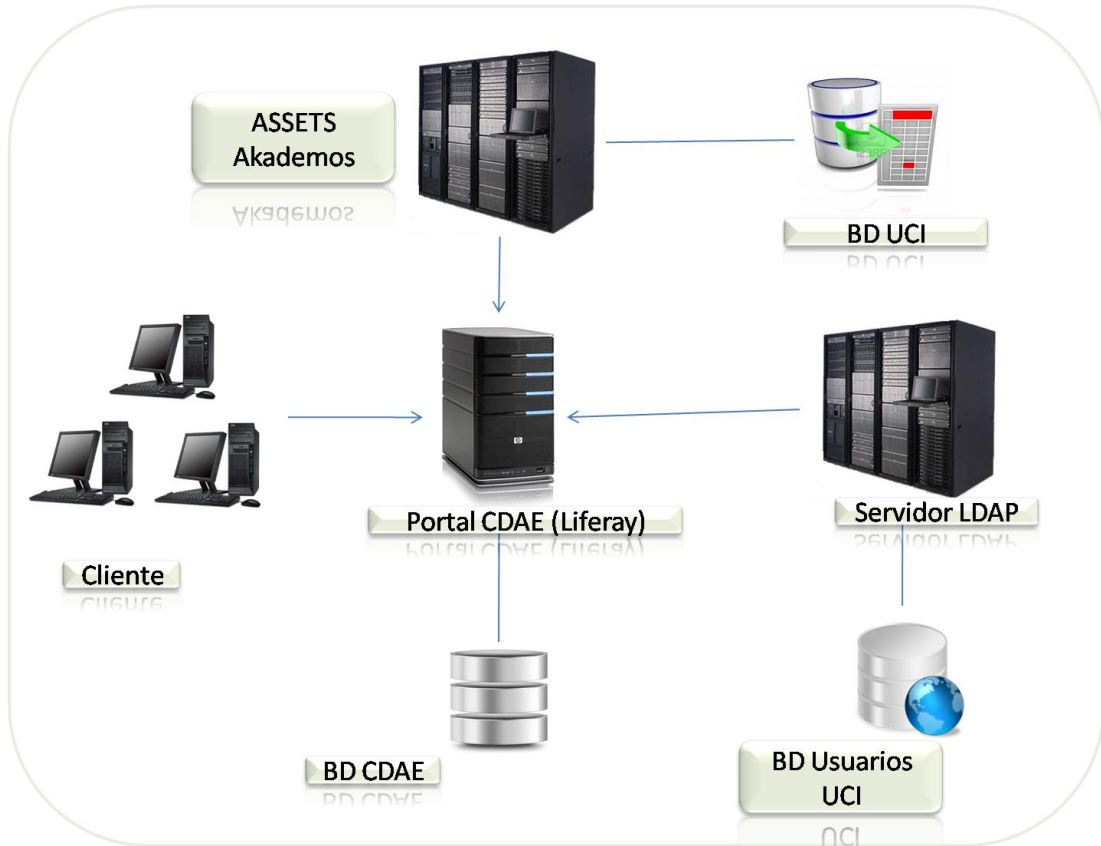


Fig. 3. Componentes propuestos para el portal CDAE.

## 2.2 Arquitectura de la Información.

### 2.2.1 Definición de la audiencia.

El portal constará de una audiencia conformada por dos grupos temáticos, los estudiantes y profesores del CDAE; y un grupo más general pero no menos importante que lo constituye primeros grupos enmarcados en un ámbito interno y el último en un marco exterior a las necesidades comunicativas e informativas del centro.

### 2.2.2 Clasificación de la audiencia.

Estudiantes:

Este grupo determina la mayoría de audiencia interna a la que va encaminado el portal, estando conformado por los estudiantes adjuntos al centro de desarrollo.

Profesores:

Se definen como los instrumentadores del manejo de información vinculada al portal. Esta audiencia está compuesta por profesores vinculados al CDAE perteneciente a la UCI.

Comunidad Universitaria:

Audiencia externa al centro la cual consumiría toda la información pública y recursos expuestos que sea de su interés.

### **2.2.3 Necesidades de la audiencia.**

Estudiantes:

El portal constituirá un medio para obtener información detallada y actualizada concerniente al desarrollo profesional de sus estudiantes así como noticias relacionadas con las tecnologías que se emplean para desarrollar en los proyectos productivos e investigaciones que se llevan a cabo en el centro y también les valdrá como un medio de aprendizaje.

Profesores:

Les proporcionará información vinculada a los diferentes departamentos que componen el centro, permitirá que sus conocimientos lleguen a la comunidad de usuarios a través de las publicaciones que realicen a cerca de sus diferentes áreas de desarrollo, enriqueciendo así el conocimiento universitario.

Comunidad Universitaria:

Proporcionará información relacionada con la línea de trabajo desempeñada por el CDAE, la comunidad podrá aprovechar todos los conocimientos publicados en el portal para así extender su horizonte profesional y enriquecer el del centro con los aportes que realice.



### 2.2.4 Mapa de navegación.

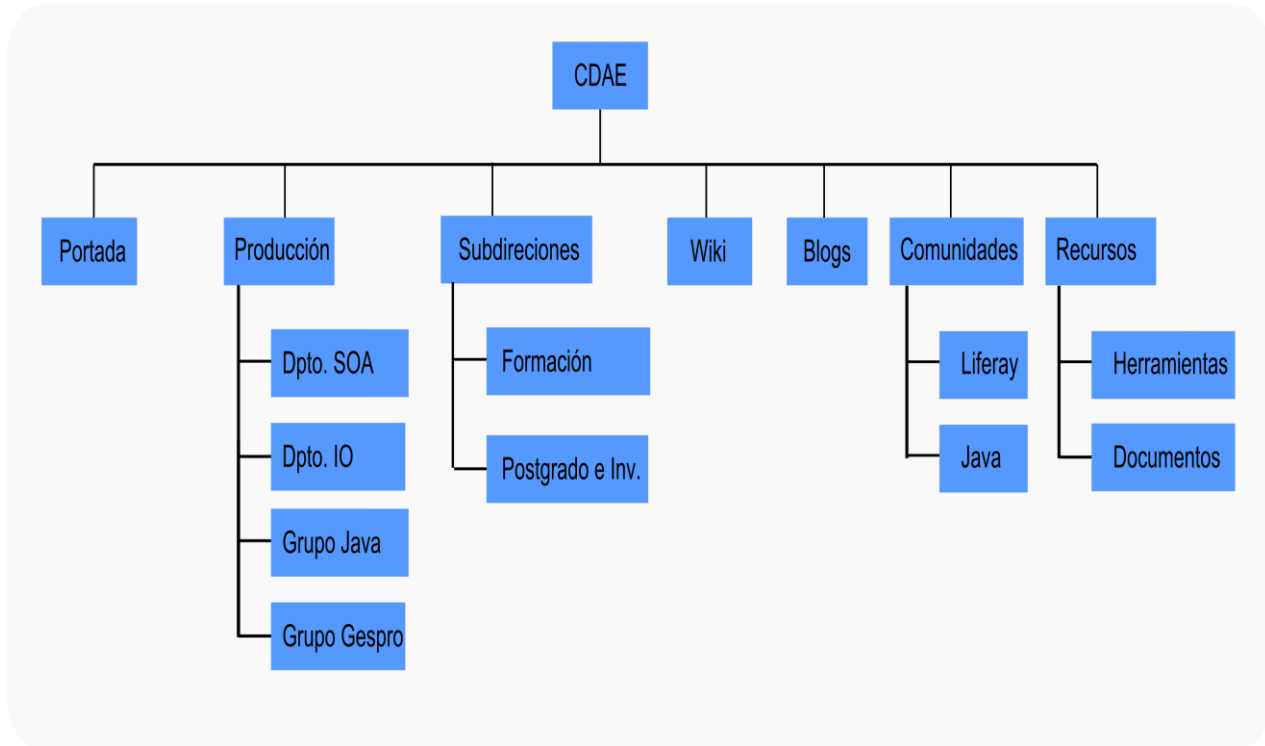


Fig. 4. Mapa de navegación, portal CDAE.

### 2.2.5 Estructura del portal.

El portal del CDAE estará conformado por un conjunto de páginas web, a las cuales tendrán acceso los distintos roles definidos. Cada página del portal estará constituida por una cabecera (header), un contenedor de portlets y un pie de página (footer). La cabecera estará ubicada en la parte superior, conformada del menú de navegación, el logotipo del centro de desarrollo en su extremo izquierdo y un menú de ruta de acceso (breadcrumbs) en su parte posterior. El contenedor de portlets en su estructura más básica constará de dos columnas, tendrá una dimensión de (30/70), aquí es donde se desplegarán los portlets que se necesiten para satisfacer las necesidades actuales del centro. El pie de página estará integrado por una serie de enlaces de interés para los internautas.

### **2.2.6 Elementos del sistema de navegación.**

- Menú General: ubicado en la parte superior de la cabecera, siempre presente en todo el sitio, permite el acceso a cada una de las áreas del sitio.
- Pie de Página: provee de un conjunto de direcciones a páginas de interés para los usuarios que visiten en el portal.
- Ruta de Acceso: listado que aparece en la parte inferior a la cabecera de cada página y que muestra el la traza o ruta de páginas que hay entre la Portada del sitio hasta la página actual que se esté revisando; cada traza debe tener un enlace, para acceder al área en la cual se encuentra la página. Cada uno de los elementos que conforman este camino debe tener un enlace que permita el acceso a esas áreas.
- Botón Inicio: Este enlace brinda la posibilidad de ir directamente a la portada desde cualquier lugar del sitio en que se encuentre el usuario.
- Mapa del sitio: Es el encargado de mostrar la estructura organizativa de las páginas que conforman el portal.
- Navegación por etiquetas: Muestra un conjunto de etiquetas relacionadas a las publicaciones realizadas en el Portal, de esta forma el usuario tendría una navegación más cómoda para satisfacer sus necesidades de información.
- Navegación por categorías: Muestra un conjunto de categorías relacionadas a las publicaciones realizadas en el Portal, de esta forma el usuario tendría una navegación más cómoda para satisfacer sus necesidades de información.
- Buscador: Presente en cada página para realizar búsquedas en todo el portal o por comunidades.

### 2.2.7 Diseño de la estructura de las pantallas tipo.

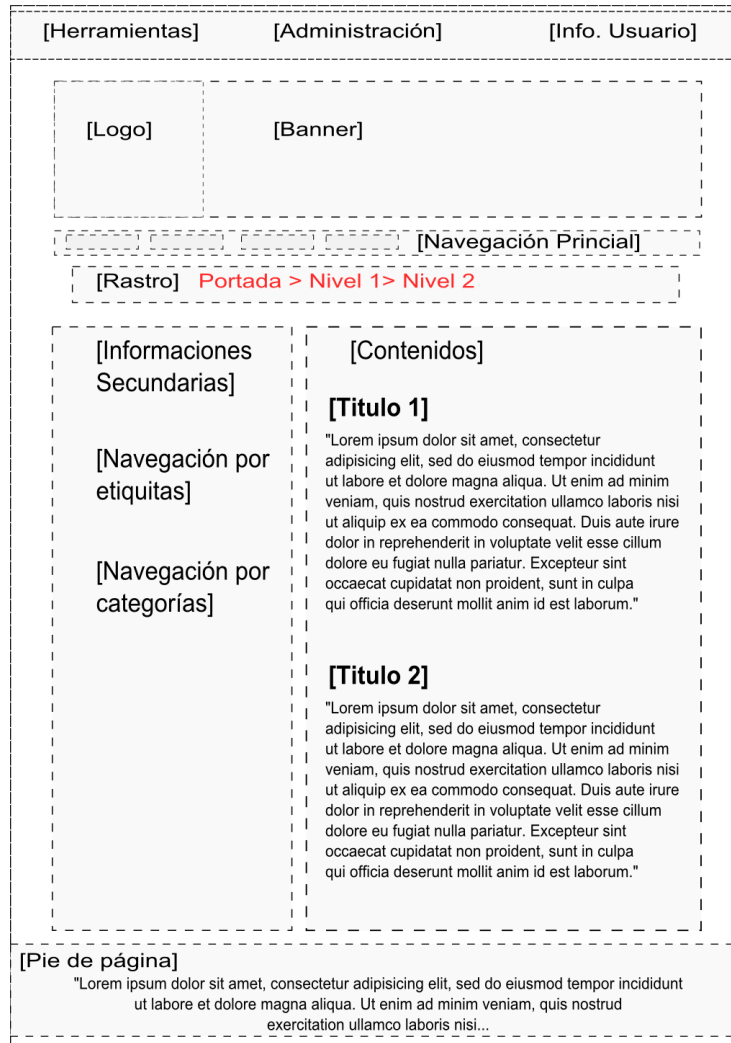


Fig. 5. Pantalla tipo.

### 2.3 Características del portal.

El portal tendrá las características de un portal empresarial proveyendo a los miembros del centro de información interna o externa relacionada con la entidad, además potenciará la participación y el intercambio a nivel universitario e interno del propio centro, para lograr este objetivo se ofrecen una serie de utilidades o aplicaciones como valor añadido, siendo los administradores los encargados de sustituir dichas aplicaciones o incorporar otras que consideren de interés.

Las aplicaciones, por defecto que se utilizarán son:

- Visor de Contenidos web
- Mapa de navegación
- Navegación por contenidos
- Navegación por etiquetas
- Foros de discusión
- Wiki
- Blog
- Chat
- Encuestas
- Nota rápida
- RSS
- Agenda

Dispondrá de un buscador de contenidos que permitirá buscar por, texto libre y comunidad, arrojando los resultados de forma inmediata, permitirá al usuario la sindicación de contenidos de sus comunidades, mediante el estándar RSS, facilitando el acceso a las fuentes de información de forma rápida y sencilla. El sistema potencia la personalización del espacio personal de cada usuario facilitando la participación del mismo para generar los contenidos que pueden ser publicados y compartidos con toda la comunidad.

El sistema contará con dos temas básicos (Fig. 5), en los que predominan los colores que identifican al centro.



Fig. 6. Temas visuales del portal.

## **2.4 Seguridad en el portal.**

Los portales web deben implementar mecanismos de seguridad para protegerse de un gran número de vulnerabilidades y amenazas, que pueden ocasionar daños en la integridad y confidencialidad de la información así como en la disponibilidad de los servicios brindados al público. Los controles que se aplican para lograr proteger la información nunca son suficientes pero si ayudan a mitigar las flaquezas en la seguridad del sistema, y los niveles de protección están en dependencia de la privacidad de los datos que se manejen en la institución.

El portal del CDAE implementa un conjunto de controles para mitigar los riesgos de seguridad a los cuales podría estar expuesto el portal CDAE. El centro puede considerar que se necesiten otros controles adicionales, debido a que los propuestos no conforman una lista exhaustiva.

No todas las aplicaciones publicadas a través del portal serán de carácter público, por tal motivo se debe proteger del acceso no autorizado a estas. El portal cuenta con secciones reservadas para el personal autorizado a acceder a dichas aplicaciones.

El portal cuenta con un mecanismo de autenticación basada en formulario Web, este mecanismo recoge las credenciales de un usuario y las confronta contra el sistema LDAP de la universidad. Se define además un control para autorizar a los usuarios con diferentes privilegios de acceso a la información, el cual se basa en los grupos y roles definidos para esos en el sistema. También presenta un gran número de aplicaciones interactivas por lo tanto su correcta validación es un tema que aporta seguridad a la integridad del portal, ya que una de las principales amenazas que enfrentan en la actualidad los portales Web es la explotación de vulnerabilidades en dichas aplicaciones web[23].

## **2.5 Metodología Extreme Programming (XP).**

### **2.5.1 Fase 1: Exploración.**

Durante esta fase se identificaron las historias de usuarios de interés para el desarrollo del producto, las cuales representan una breve descripción del comportamiento del sistema. Solo se describen en este acápite cuatro de las dieciocho historias de usuarios, las restantes podrán ser consultadas en el [Anexo 3](#).

**Historias de Usuario.**

<b>Historia de Usuario</b>	
<b>Nombre:</b> Importar estudiante	
<b>Número:</b> 1	<b>Usuario:</b> Desarrollador
<b>Dependiente:</b> No	
<b>Prioridad:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Descripción:</b> Esta funcionalidad será la encargada de registrar a los estudiantes vinculados al centro en el sistema. Todos los datos asociados a los estudiantes serán importados mediante el consumo de servicios del Sistema de Gestión Académica de la Universidad (Akademos).	

**HU 1. Importar estudiante.**

<b>Historia de Usuario</b>	
<b>Nombre:</b> Buscar estudiante	
<b>Número:</b> 2	<b>Usuario:</b> Desarrollador
<b>Dependiente:</b> 1	
<b>Prioridad:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Descripción:</b> Esta funcionalidad será la encargada de filtrar según el criterio que desee el usuario los estudiantes vinculados al centro que se encuentren registrados en el sistema o estén siendo importados a este.	

**HU 2. Buscar estudiante.**

<b>Historia de Usuario</b>	
<b>Nombre:</b> Modificar estudiante	
<b>Número:</b> 3	<b>Usuario:</b> Desarrollador
<b>Dependiente:</b> 1	
<b>Prioridad:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Descripción:</b> Esta funcionalidad será la encargada de agregar o cambiar algunos datos de interés de los estudiantes vinculados al centro que se encuentren registrados en el sistema o estén siendo importados a este.	

**HU 3. Modificar estudiante.**

<b>Historia de Usuario</b>	
<b>Nombre:</b> Registrar tema de tesis	
<b>Número:</b> 4	<b>Usuario:</b> Desarrollador
<b>Dependiente:</b> No	
<b>Prioridad:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Descripción:</b> Esta funcionalidad será la encargada de registrar los temas de tesis propuestos por los profesores pertenecientes al centro.	

**HU 4. Registrar tema de tesis.****2.5.2 Fase 2: Planificación de la Entrega.**

En esta fase se establece la prioridad de cada historia de usuario, y se realiza una estimación del esfuerzo necesario para implementar cada una de ellas. Esto se expresa utilizando como unidad medida el punto, el cual se consideró como 4 días de trabajo en programación real.

**Estimación de esfuerzo por Historias de Usuario.**

No	Historia de Usuario	Puntos de Estimación
1	Importar estudiante	1
2	Buscar estudiante	0.5
3	Modificar estudiante	0.5
4	Registrar tema de tesis	1
5	Buscar tema de tesis	1
6	Modificar tema de tesis	1
7	Aprobar tema de tesis	1
8	Solicitud de tema de tesis	1
9	Asignar tema de tesis	1
10	Crear tribunal de tesis	0.5
11	Consultar tribunal de tesis	0.5
12	Asignar tesis a un tribunal	2
13	Evaluar cortes de tesis	2
14	Asignar evaluaciones periódicas	1
15	Consultar evaluaciones periódicas	2
16	Importar profesor	1
17	Modificar profesor	0.5
18	Buscar profesor	0.5

**Tabla 6. Estimación de esfuerzo por Historias de Usuario.****2.5.3 Fase 3: Iteraciones.**

En esta fase se definieron tres iteraciones para la realización del sistema:

Iteración 1



Las HU que presentan mayor prioridad serán implementadas en esta primera iteración, dando al sistema las primeras funcionalidades, importar estudiantes y profesores así como modificar los datos de estudiantes y profesores.

#### Iteración 2

En la segunda iteración se implementarán las HU relacionadas con registrar, modificar, aprobar, solicitar, asignar y buscar tema de tesis.

#### Iteración 3

Luego de haberse desarrollado las funcionalidades de alta prioridad para el sistema se elaboran las HU crear y buscar tribunal de tesis, asignar tesis a un tribunal, evaluar cortes de tesis, asignar evaluaciones periódicas y consultarlas.

#### Plan de duración de las iteraciones.

Cantidad de Iteraciones	Orden de las Historia de Usuario a implementar	Duración total de las iteraciones
1	Importar estudiante	4 días
1	Buscar estudiante	2 días
1	Modificar estudiante	2 días
1	Importar profesor	4 días
1	Modificar profesor	2 días
1	Buscar profesor	2 días
2	Registrar tema de tesis	4 días
2	Buscar tema de tesis	4 días
2	Modificar tema de tesis	4 días
2	Aprobar tema de tesis	4 días
2	Solicitud de tema de tesis	4 días

2	Asignar tema de tesis	4 días
3	Crear tribunal de tesis	2 días
3	Consultar tribunal de tesis	8 días
3	Asignar tesis a un tribunal	8 días
3	Evaluar cortes de tesis	8 días
3	Asignar evaluaciones periódicas	4 días
3	Consultar evaluaciones periódicas	8 días

**Tabla 7. Plan de duración de las iteraciones.**

### Plan de Entregas.

Historia de Usuario	Final 1ra iteración	Final 2ra iteración	Final 3ra iteración
Importar estudiante	x		
Buscar estudiante	x		
Modificar estudiante	x		
Importar profesor	x		
Modificar profesor	x		
Buscar profesor	x		
Registrar tema de tesis		x	
Buscar tema de tesis		x	
Modificar tema de tesis		x	
Aprobar tema de tesis		x	
Solicitar tema de tesis		x	
Asignar tema de tesis		x	
Crear tribunal de tesis			x
Consultar tribunal de tesis			x
Asignar tesis a un tribunal			x

Evaluar cortes de tesis			x
Asignar evaluaciones periódicas			x
Consultar evaluaciones periódicas			x

**Tabla 8. Plan de entregas.**

### Plan de Tareas.

Se descompone cada HU en varias tareas que serán desarrolladas para dar cumplimiento a la misma, de esta forma se facilita el trabajo de implementación de los programadores.

Historias de Usuario	Tareas
Importar estudiante	1 Definir una estructura de entidades como soporte para la información solicitada. 2 Definir y crear los componentes visuales necesarios para almacenar la información a almacenar y ejecutar el procedimiento de almacenamiento de los estudiantes.
Buscar estudiante	1 Definir y crear los componentes visuales necesarios para mostrar los resultados arrojados según el criterio de búsqueda del usuario la información deseada por el usuario.
Modificar estudiante	1 Definir y crear los componentes visuales necesarios para modificar la información deseada por el usuario y ejecutar el procedimiento de actualización de datos de los estudiantes.
Importar profesor	1 Definir una estructura de entidades como soporte para la información solicitada. 2 Definir y crear los componentes visuales necesarios para almacenar la información a

	almacenar y ejecutar el procedimiento de almacenado de los profesores.
Buscar Profesor	1 Definir y crear los componentes visuales necesarios para mostrar los resultados arrojados según el criterio de búsqueda del usuario la información deseada por el usuario.
Modificar profesor	1 Definir y crear los componentes visuales necesarios para modificar la información deseada por el usuario y ejecutar el procedimiento de actualización de datos de los profesores.
Registrar tema de tesis	1 Definir una estructura de entidades como soporte para la información solicitada. 2 Definir y crear los componentes visuales necesarios para almacenar la información a almacenar y ejecutar el procedimiento de almacenado de las Tesis.
Modificar tema de tesis	1 Definir y crear los componentes visuales necesarios para modificar la información deseada por el usuario y ejecutar el procedimiento de actualización de datos de las Tesis.
Buscar tema de tesis	1 Definir y crear los componentes visuales necesarios para mostrar los datos obtenidos según la petición del usuario y ejecutar el procedimiento de búsqueda de Tesis.
Aprobar tema de tesis	1 Definir y crear los componentes visuales necesarios para mostrar la información

	relacionada con el tema de tesis.
Solicitar tema de tesis	1 Definir y crear los componentes visuales necesarios para mostrar la información relacionada con los temas de tesis aprobados por el centro.
Crear tribunal de tesis	1 Definir una estructura de entidades como soporte para la información solicitada. 2 Definir y crear los componentes visuales necesarios para almacenar la información relacionada con el tribunal y ejecutar el procedimiento de almacenado del tribunal de tesis.
Consultar tribunal de tesis	1 Definir y crear los componentes visuales necesarios para mostrar los datos obtenidos según la petición del usuario y ejecutar el procedimiento de búsqueda del tribunal de tesis.
Asignar tesis a un tribunal	1 Definir una estructura de entidades como soporte para la información solicitada. 2 Definir y crear los componentes visuales necesarios para almacenar la información relacionada con la tesis y el tribunal; ejecutar el procedimiento de almacenado.
Evaluar cortes de tesis	1 Definir y crear los componentes visuales necesarios para almacenar la información relacionada con los cortes realizados y ejecutar el procedimiento de almacenado.
Asignar evaluaciones periódicas	1 Definir una estructura de entidades como soporte para la información solicitada. 2 Definir y crear los componentes visuales

	necesarios para almacenar la información a almacenar y ejecutar el procedimiento de almacenado de la evaluaciones.
Consultar evaluaciones periódicas	2 Definir y crear los componentes visuales necesarios para visualizar la información almacenada concerniente a las evaluaciones periódicas de los alumnos del CDAE.

Tabla 9. Plan de tareas.

### 2.5.4 Fase 4: Producción.

En esta fase se definen las clases del sistema (Fig. 7), su tipo y la relación entre ellas, con todas sus funciones (atributos y métodos).

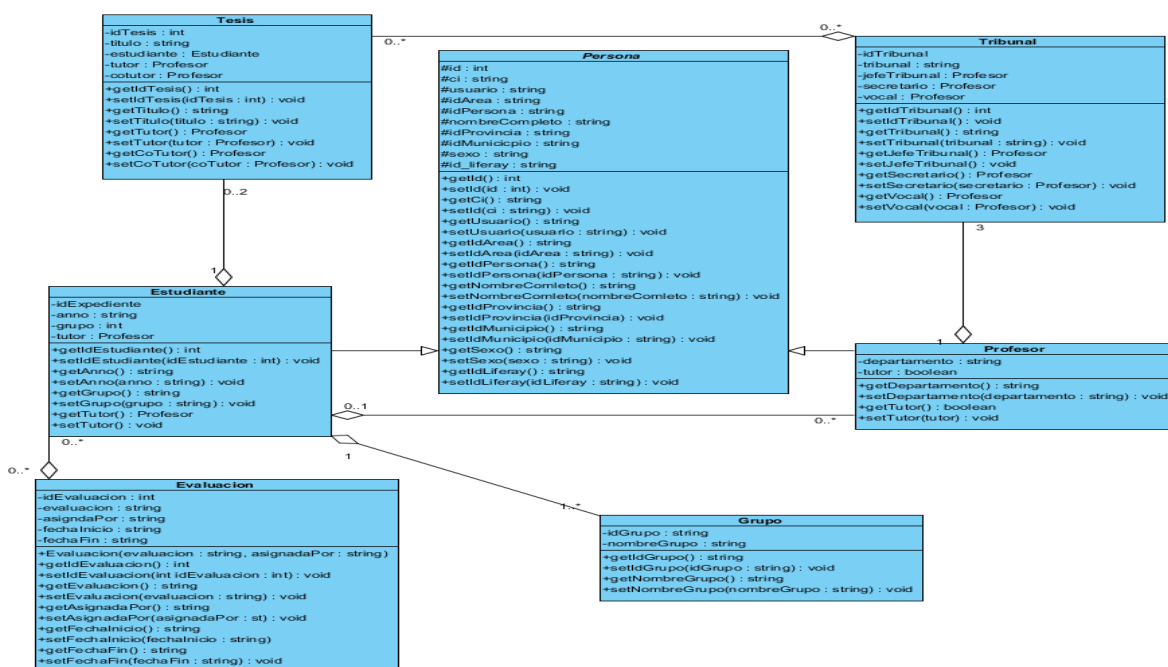


Fig. 8. Clases del Sistema.

### Descripción de las Clases.

Tribunal	
Responsabilidades	Tipo de clase
-Esta clase contiene la información necesaria para generar un nuevo tribunal. - Permite cambiarle el valor inicial de uno o varios atributos a un tribunal -Permite devolver el valor de cualquier atributo de un tribunal.	Modelo

**Tarjeta CRC 1. Calse Tribunal.**

TribunalDAO	
Responsabilidades	Colaboradores
Esta clase es responsable de: -Adicionar un nuevo tribunal. -Modificarle uno o varios atributos a un tribunal determinado. -Buscar uno o varios tribunales en específico. -Eliminar un tribunal. -Listar los tribunales existentes.	Profesor Tesis

**Tarjeta CRC 2. Clase TribunalDAO.**

Tesis	
Responsabilidades	Tipo de clase
-Esta clase contiene la información necesaria para generar un nuevo tema de tesis. - Permite cambiarle el valor inicial de uno o varios atributos a un tema. -Permite devolver el valor de cualquier atributo de un tema.	Modelo

**Tarjeta CRC 3. Clase Tesis.**

TesisDAO	
Responsabilidades	Colaboradores
Esta clase es responsable de: -Adicionar un nuevo tema de tesis. -Modificarle uno o varios atributos a un tema de tesis. -Buscar una o varios temas tesis en específico. -Eliminar un tema de tesis.	Profesor Estudiante

**Tarjeta CRC 4. Clase TesisDAO.**

Profesor	
Responsabilidades	Tipo de clase
-Esta clase contiene la información necesaria para generar un nuevo profesor. - Permite cambiarle el valor inicial de uno o varios atributos a un profesor. -Permite devolver el valor de cualquier atributo de un profesor.	Modelo

**Tarjeta CRC 5. Clase Profesor.**

ProfesorDAO	
Responsabilidades	Colaboradores
Esta clase es responsable de: -Adicionar un nuevo profesor. -Modificarle uno o varios atributos a un profesor determinado. -Buscar uno o varios profesores en específico. -Eliminar un profesor.	-

**Tarjeta CRC 6. Clase ProfesorDAO.**

Estudiante	
Responsabilidades	Tipo de clase



<ul style="list-style-type: none"> <li>-Esta clase contiene la información necesaria para generar un nuevo estudiante.</li> <li>- Permite cambiarle el valor inicial de uno o varios atributos a un estudiante.</li> <li>-Permite devolver el valor de cualquier atributo de un estudiante.</li> </ul>	Modelo
--	--------

**Tarjeta CRC 7. Clase Estudiante.**

EstudianteDAO	
Responsabilidades	Colaboradores
Esta clase es responsable de: <ul style="list-style-type: none"> <li>-Adicionar un nuevo estudiante.</li> <li>-Modificarle uno o varios atributos a un estudiante determinado.</li> <li>-Buscar uno o varios estudiantes en específico.</li> <li>-Eliminar un estudiante.</li> </ul>	-

**Tarjeta CRC 8. Clase EstudianteDAO.**

Grupo	
Responsabilidades	Tipo de clase
<ul style="list-style-type: none"> <li>-Esta clase contiene la información necesaria para generar un nuevo grupo.</li> <li>- Permite cambiarle el valor inicial de uno o varios atributos a un grupo.</li> <li>-Permite devolver el valor de cualquier atributo de un grupo.</li> </ul>	Modelo

**Tarjeta CRC 9. Clase Grupo.**

GrupoDAO	
Responsabilidades	Colaboradores
Esta clase es responsable de:	Grupo(Servicio web)

-Adicionar un nuevo grupo. -Modificarle uno o varios atributos a un grupo determinado. -Buscar uno o varios grupos en específico. -Eliminar un grupo.	
--	--

Tarjeta CRC 10. Clase GrupoDAO.

### Diseño de la Base de Datos.

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener la base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, etc.[24].

### Diagrama de clases persistentes (Fig. 9).

La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Algunas de las reglas que se recomiendan aplicar son:

- Cuando una clase está formada por otras clases es persistente, automáticamente las que la componen también son persistentes. Lo contrario no se cumple necesariamente.
- Cuando una clase hija de una jerarquía es persistente, automáticamente son persistentes sus ancestros en el árbol de jerarquía. Lo contrario no se cumple necesariamente.
- Cuando se define como persistente a una clase que agrupa a objetos de un mismo tipo de la clase base (listas, colecciones, registros), entonces son persistentes todas las clases hijas a partir de la clase base, incluyéndose la clases base.
- Cuando hay herencia múltiple, esta debe ser resuelta antes si el medio de almacenamiento no soporta este concepto[25].

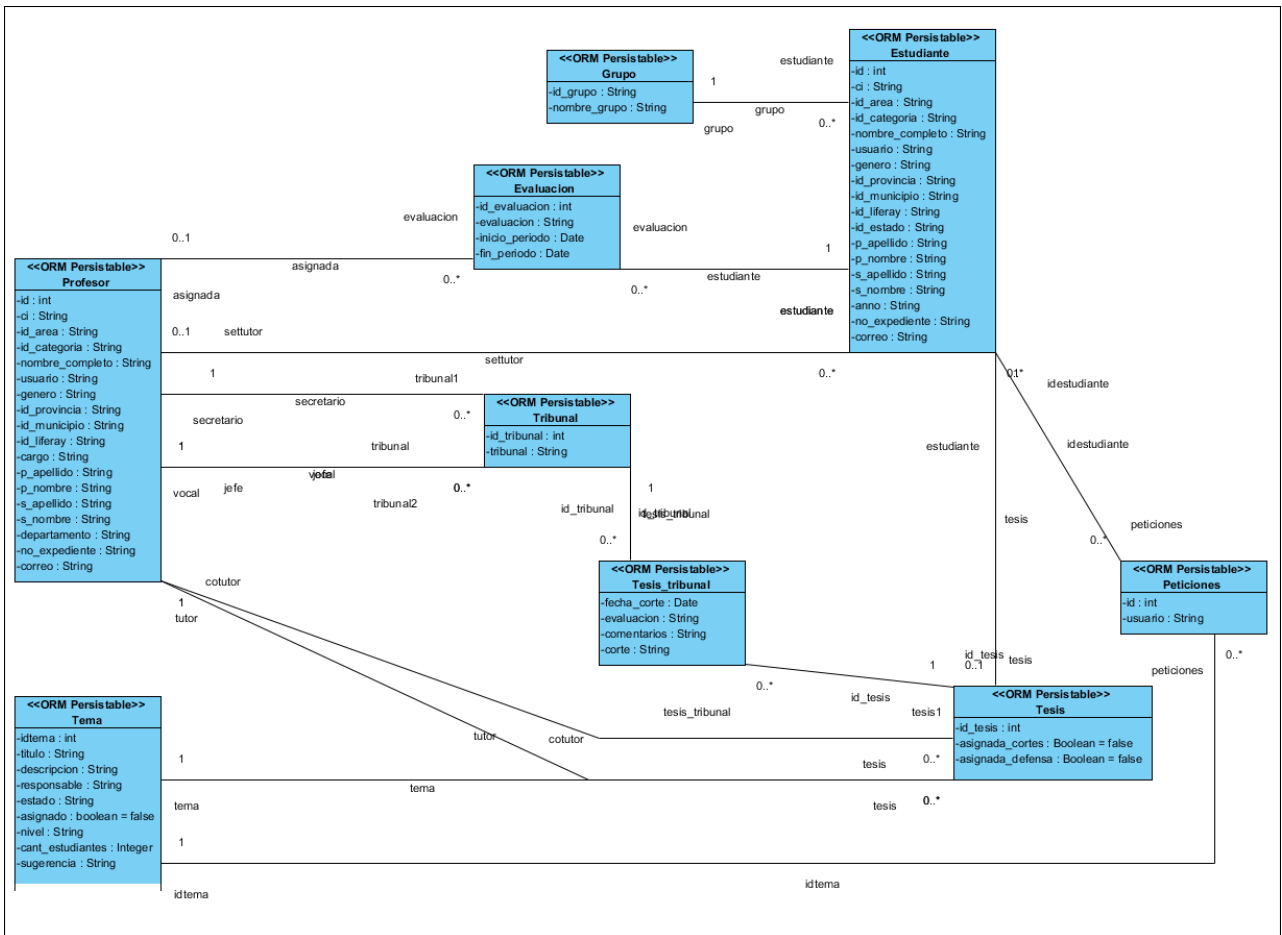


Fig. 10. Clases Persistentes.

### Diagrama Entidad-Relación (Fig. 11).

El modelo Entidad-Relación es uno de los enfoques de modelización de datos que más se utiliza actualmente por su simplicidad y legibilidad. Su legibilidad se ve favorecida porque proporciona una notación diagramática muy comprensiva. Es una herramienta útil tanto para ayudar al diseñador a reflejar en un modelo conceptual los requisitos del mundo real de interés como para comunicarse con el usuario final sobre el modelo conceptual obtenido y, de este modo, poder verificar si satisface sus requisitos[24].

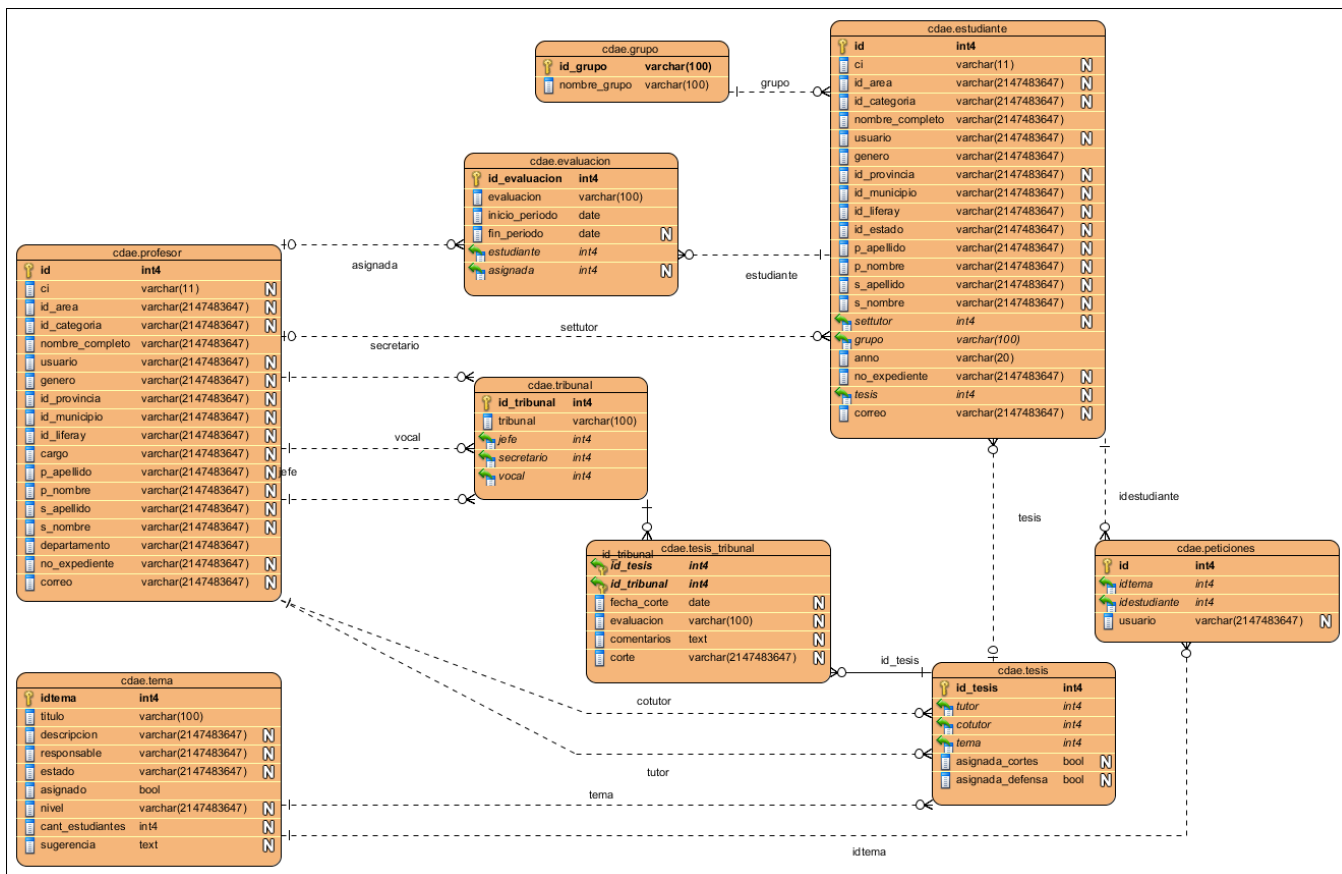


Fig. 12. Diagrama Entidad-Relación.

## 2.6. Conclusiones del capítulo.

El desarrollo de este capítulo facilita el análisis y comprensión del tema propuesto. Se describen las tecnologías y las herramientas a empleadas en el proceso de desarrollo del portal, la metodología que guía la construcción de la solución y la arquitectura a emplear en el sistema, la cual posibilita que el portal sea fácilmente corregible a integrable con otros sistemas.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

**3.1 Introducción.**

En el presente capítulo se lleva a cabo la descripción de las tareas de implementación generadas por cada historia de usuario realizadas en las tres iteraciones propuestas, obteniendo en cada una de ellas una versión del producto, así como las pruebas de aceptación efectuadas sobre el portal.

**3.2 Implementación.****3.2.1 Iteración 1.**

En esta iteración se desarrollan las HU que presentan mayor prioridad, dando al sistema las primeras funcionalidades, estas son: importar estudiantes y profesores así como modificar los datos de estudiantes y profesores. Solo serán definidas en esta sección las tareas realizadas para la primera iteración las restantes iteraciones podrán ser consultadas en el [Anexo 4](#) y el [Anexo 5](#).

**Historias de Usuario implementadas en la primera iteración.**

Historias de Usuarios	Estimación	Real
Importar estudiante	1	1
Buscar estudiante	0.5	0.5
Modificar estudiante	0.5	0.5
Importar professor	1	1
Buscar profesor	0.5	0.5
Modificar professor	0.5	0.5

**Tareas de las historias de usuario implementadas en la primera iteración.**

Tareas implementadas de la historia de usuario "Importar estudiante". [Ver Anexo.](#)

Número tarea: 1	Historia de Usuario # 1
<b>Nombre tarea:</b> Definición y desarrollo de la interfaz	
Importar estudiante.	

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 27/02/2012	<b>Fecha fin:</b> 28/02/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza el diseño de la pantalla modificar profesor. Se crea el diseño realizando una correspondencia entre los tipos de datos a entrar, los datos soportados y los componentes visuales que más se adapten a estos.	

<b>Número tarea: 2</b>	<b>Historia de Usuario # 1</b>
<b>Nombre tarea:</b> Validación de la entrada de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 28/02/2012	<b>Fecha fin:</b> 29/02/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza la validación de datos, asegurando así el correcto funcionamiento de la aplicación. Para ello se tiene presente los tipos de datos soportados y los posibles a entrar por los usuarios, así como los campos obligatorios a llenar.	

<b>Número tarea: 3</b>	<b>Historia de Usuario # 1</b>
<b>Nombre tarea:</b> Conexión con la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 29/02/2012	<b>Fecha fin:</b> 01/03/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza la conexión con la base de datos definida para el portal, haciendo uso de una Clase HibernateUtil.	

<b>Número tarea: 4</b>	<b>Historia de Usuario # 1</b>
<b>Nombre tarea:</b> Prueba de la aplicación importar estudiante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 01/3/2012	<b>Fecha fin:</b> 01/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se le realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar esta tarea.	

Tareas implementadas de la historia de usuario “Buscar estudiante”.

<b>Número tarea: 1</b>	<b>Historia de Usuario # 2</b>
<b>Nombre tarea:</b> Diseño de la pantalla para buscar estudiantes.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 02/3/2012	<b>Fecha fin:</b> 03/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza el diseño de la pantalla buscar estudiantes, la cual presenta los criterios por los cuales se puede filtrar un listado de estudiantes.	

Tareas implementadas de la historia de usuario “Modificar estudiante”.

<b>Número tarea: 1</b>	<b>Historia de Usuario # 3</b>
<b>Nombre tarea:</b> Definición y desarrollo de la interfaz modificar estudiante.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.10
<b>Fecha inicio:</b> 03/3/2012	<b>Fecha fin:</b> 03/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	

<p><b>Descripción:</b></p> <p>Se realiza el diseño de la pantalla modificar profesor. Se crea el diseño realizando una correspondencia entre los tipos de datos a entrar, los datos soportados y los componentes visuales que más se adapten a estos.</p>
---

<b>Número tarea: 2</b>	<b>Historia de Usuario # 3</b>
<b>Nombre tarea:</b> Validación de la entrada de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha inicio:</b> 03/3/2012	<b>Fecha fin:</b> 03/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<p><b>Descripción:</b></p> <p>Se realiza la validación de datos, asegurando así el correcto funcionamiento de la aplicación. Para ello se tiene presente los tipos de datos soportados y los posibles a entrar por los usuarios, así como los campos obligatorios a llenar.</p>	

<b>Número tarea: 3</b>	<b>Historia de Usuario # 3</b>
<b>Nombre tarea:</b> Conexión con la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.13
<b>Fecha inicio:</b> 04/3/2012	<b>Fecha fin:</b> 04/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<p><b>Descripción:</b></p> <p>Se realiza la conexión con la base de datos definida para el portal, haciendo uso de una Clase HibernateUtil.</p>	

<b>Número tarea: 4</b>	<b>Historia de Usuario # 3</b>
<b>Nombre tarea:</b> Prueba de la aplicación modificar estudiante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.13



<b>Fecha inicio:</b> 04/3/2012	<b>Fecha fin:</b> 04/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se le realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar esta tarea. Para ello se le introducen una serie de datos (correctos e incorrectos), para así comparar los resultados devueltos y los esperados.	

Tareas implementadas de la historia de usuario "Importar profesor". [Ver Anexo.](#)

<b>Número tarea: 1</b>	<b>Historia de Usuario # 16</b>
<b>Nombre tarea:</b> Definición y desarrollo de la interfaz importar profesor.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 05/3/2012	<b>Fecha fin:</b> 05/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza el diseño de la pantalla modificar profesor. Se crea el diseño realizando una correspondencia entre los tipos de datos a entrar, los datos soportados y los componentes visuales que más se adapten a estos.	

<b>Número tarea: 2</b>	<b>Historia de Usuario # 16</b>
<b>Nombre tarea:</b> Validación de la entrada de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 06/3/2012	<b>Fecha fin:</b> 06/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza la validación de datos, asegurando así el correcto funcionamiento de la aplicación. Para ello se tiene presente los	

tipos de datos soportados y los posibles a entrar por los usuarios, así como los campos obligatorios a llenar.

<b>Número tarea: 3</b>	<b>Historia de Usuario # 16</b>
<b>Nombre tarea:</b> Conexión con la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 07/3/2012	<b>Fecha fin:</b> 07/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza la conexión con la base de datos definida para el portal, haciendo uso de una Clase HibernateUtil.	

<b>Número tarea: 4</b>	<b>Historia de Usuario # 16</b>
<b>Nombre tarea:</b> Prueba de la aplicación importar profesor.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.25
<b>Fecha inicio:</b> 07/3/2012	<b>Fecha fin:</b> 08/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se le realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar esta tarea.	

Tareas implementadas de la historia de usuario "Buscar profesor".

<b>Número tarea: 1</b>	<b>Historia de Usuario # 18</b>
<b>Nombre tarea:</b> Diseño de la pantalla para buscar estudiantes.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.50
<b>Fecha inicio:</b> 09/3/2012	<b>Fecha fin:</b> 10/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	

**Descripción:**

Se realiza el diseño de la pantalla buscar profesor, la cual presenta los criterios por los cuales se puede filtrar un listado de profesores.

Tareas implementadas de la historia de usuario “Modificar profesor”.

Número tarea: 1	Historia de Usuario # 17
<b>Nombre tarea:</b> Definición y desarrollo de la interfaz modificar profesor.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.10
<b>Fecha inicio:</b> 11/3/2012	<b>Fecha fin:</b> 11/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b>	
Se realiza el diseño de la pantalla modificar profesor. Se crea el diseño realizando una correspondencia entre los tipos de datos a entrar, los datos soportados y los componentes visuales que más se adapten a estos.	

Número tarea: 2	Historia de Usuario # 17
<b>Nombre tarea:</b> Validación de la entrada de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.14
<b>Fecha inicio:</b> 11/3/2012	<b>Fecha fin:</b> 11/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b>	
Se realiza la validación de datos, asegurando así el correcto funcionamiento de la aplicación. Para ello se tiene presente los tipos de datos soportados y los posibles a entrar por los usuarios, así como los campos obligatorios a llenar.	

<b>Número tarea: 3</b>	<b>Historia de Usuario # 17</b>
<b>Nombre tarea:</b> Conexión con la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.13
<b>Fecha inicio:</b> 12/3/2012	<b>Fecha fin:</b> 12/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se realiza la conexión con la base de datos definida para el portal, haciendo uso de una Clase HibernateUtil.	

<b>Número tarea: 4</b>	<b>Historia de Usuario # 17</b>
<b>Nombre tarea:</b> Prueba de la aplicación modificar profesor.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.13
<b>Fecha inicio:</b> 12/3/2012	<b>Fecha fin:</b> 12/3/2012
<b>Programador responsable:</b> Armando Masó Mosqueda	
<b>Descripción:</b> Se le realizan una serie de pruebas funcionales a la aplicación, con la finalidad de obtener posibles fallas al realizar esta tarea. Para ello se le introducen una serie de datos (correctos e incorrectos), para así comparar los resultados devueltos y los esperados.	

### 3.3 Pruebas.

El desarrollo de software implica la realización de una serie de actividades entre las que se encuentra la de pruebas, siendo esta una actividad que garantiza la calidad del producto desarrollado.

La metodología XP anima a probar constantemente tanto como sea posible, siendo las pruebas uno de sus pilares fundamentales. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

El sistema de programación extrema divide las pruebas en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

Las pruebas unitarias realizan un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, comprobando los bucles y condiciones, y examinado el estado del programa en varios puntos. Estas nos ayudan a tener los productos casi 100% seguros, pero supone un estudio demasiado absoluto, que ampliaría los planes de desarrollo del portal, por lo que solo se realizará un estudio de las pruebas funcionales del sistema y de su rendimiento.

### 3.3.1 Pruebas de aceptación.

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Son consideradas como “pruebas de caja negra” (*“Black box system tests”*). Se definen para cada historia de usuario y los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación[26].

Estas pruebas tienen una importancia crítica para el éxito de una iteración. Por lo tanto el encargado de las pruebas (tester) debe tenerlas listas lo antes posible a partir del comienzo de la iteración, y lograr que el cliente las apruebe para poder presentárselas cuanto antes al equipo de desarrollo[27].

#### Pruebas de aceptación al sistema.

A continuación se muestran algunos casos de pruebas relacionados con las historias de usuarios importar, modificar y buscar estudiantes. Las restantes podrán ser consultadas en los anexos. [Ver Anexo 5.](#)

Prueba de aceptación	
<b>Código:</b> HU1-P1	<b>HU:</b> #1
<b>Nombre:</b> Importar estudiantes	

<p><b>Descripción:</b></p> <p>Esta funcionalidad será la encargada de registrar a los estudiantes vinculados al centro en el sistema. Todos los datos asociados a los estudiantes serán importados mediante el consumo de servicios del Sistema de Gestión Académica de la Universidad (Akademos).</p>
<p><b>Condiciones de ejecución:</b></p> <p>El usuario debe de estar autenticado y tener privilegios de administrador para tener acceso a este servicio.</p>
<p><b>Entrada/Pasos de ejecución:</b></p> <p>Se selecciona del listado proporcionado por la aplicación la facultad y en consecuencia se selecciona el grupo al que pertenecen los estudiantes. Luego se pulsa el botón importar y los datos son almacenados en la base de datos del portal.</p>
<p><b>Resultado esperado:</b></p> <p>Mostrar un mensaje indicando el éxito de la operación.</p>
<p><b>Resultado obtenido:</b></p> <p>Se muestra un mensaje indicando el éxito de la operación realizada.</p>
<p><b>Evaluación de la prueba:</b></p> <p>Satisfactoria.</p>

**Prueba de aceptación #1**

<b>Prueba de aceptación</b>	
<b>Código: HU2-P1</b>	<b>HU: #2</b>
<b>Nombre:</b> Buscar estudiante	
<p><b>Descripción:</b></p> <p>Esta funcionalidad será la encargada de filtrar según el criterio que desee el usuario los estudiantes vinculados al centro que se encuentren registrados en el sistema o estén siendo importados a este.</p>	
<b>Condiciones de ejecución:</b>	

El usuario debe de estar autenticado y tener privilegios de administrador para tener acceso a este servicio.
<b>Entrada/Pasos de ejecución:</b> Se escribe la cadena por la que se desea filtrar los datos.
<b>Resultado esperado:</b> Se obtiene la respuesta de forma inmediata.
<b>Resultado obtenido:</b> Se obtiene la respuesta de forma inmediata.
<b>Evaluación de la prueba:</b> Satisfactoria.

**Prueba de aceptación #2**

Prueba de aceptación	
<b>Código:</b> HU3-P1	<b>HU:</b> #3
<b>Nombre:</b> Modificar estudiante	
<b>Descripción:</b> Esta funcionalidad será la encargada de agregar o cambiar algunos datos de interés de los estudiantes vinculados al centro que se encuentren registrados en el sistema o estén siendo importados a este.	
<b>Condiciones de ejecución:</b> El usuario debe de estar autenticado y tener privilegios de administración para tener acceso a este servicio.	
<b>Entrada/Pasos de ejecución:</b> 1- Se entran los datos a modificar del estudiante seleccionado (correo y usuario) correctamente. 2- Se entran los datos a modificar del estudiante seleccionado (correo y usuario) de forma incorrecta.	
<b>Resultado esperado:</b> 1- El sistema muestra un mensaje indicando el éxito de la operación.	

2- El sistema muestra un mensaje indicando el intento fallido de la operación.
<b>Resultado obtenido:</b>
1- El sistema muestra un mensaje indicando el éxito de la operación.
2- El sistema muestra un mensaje indicando el intento fallido de la operación.
<b>Evaluación de la prueba:</b>
Satisfactoria.

### Prueba de aceptación #3

#### 3.3.2 Pruebas de rendimiento.

Dado que la metodología propone las pruebas de aceptación y las unitarias, se hace necesario realizar las pruebas de rendimiento debido al tipo de software que se desarrolla.

Las pruebas de rendimiento son aquellas que son realizadas para determinar qué tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos[28].

Estas pruebas son utilizadas con diferentes propósitos; como demostrar que el sistema cumple los criterios de rendimiento establecidos, comparar dos sistemas para encontrar cuál presenta un mejor funcionamiento o medir que partes del sistema muestra un bajo rendimiento son ejemplos del uso que se les puede dar a este tipo de pruebas. El uso de herramientas que permitan el monitoreo y diagnóstico del producto es de gran importancia para realizar tareas de este tipo.

#### Pruebas de rendimiento al sistema.

Características del ambiente de pruebas.

<b>Hardware</b>	
<b>Tipo de Procesador</b>	Core Duo
<b>Memoria del Sistema</b>	4g RAM
<b>Tipo de Red</b>	LAN
<b>Software</b>	



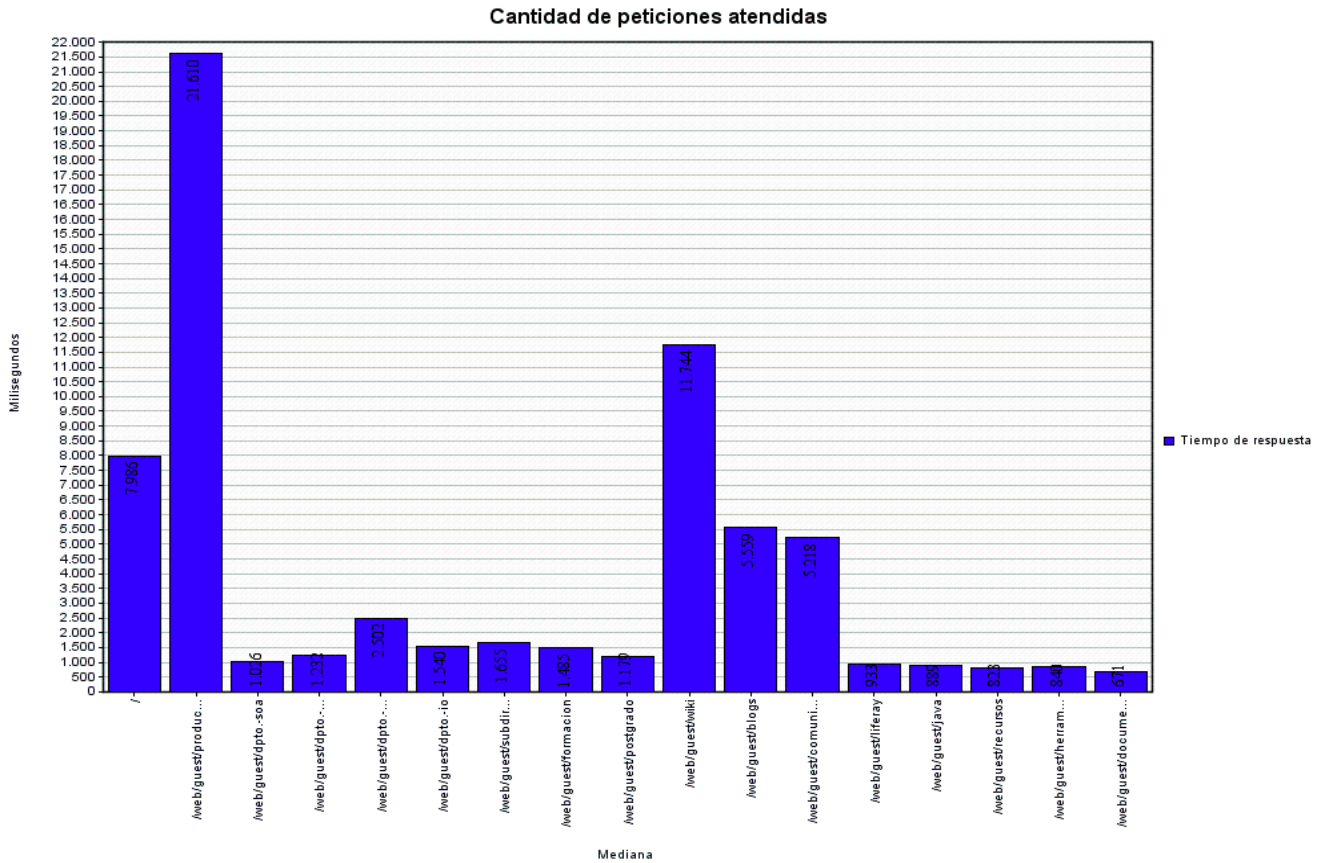
Tipo de Servidor web	Apache Tomcat
Memoria máxima	4g RAM
Máximo de hilos concurrentes	50/páginas
Plataforma	Centos Linux 6.2
Servidor de base de datos	PostgreSQL 8.4

### Resultados obtenidos.

Para el ambiente de trabajo antes descrito se obtuvieron los siguientes resultados:

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/	50	7842	7986	9049	5506	9688	0,00%	5,0/sec	151,5
/web/guest/produccion.jsessio...	50	22322	21610	27123	17038	27993	0,00%	1,6/sec	38,5
/web/guest/dpto.-soa	50	1404	1026	2894	179	5710	0,00%	2,8/sec	32,5
/web/guest/dpto.-java	50	1695	1232	3125	142	5134	0,00%	2,3/sec	26,9
/web/guest/dpto.-gespro	50	2855	2502	5346	485	7135	0,00%	2,1/sec	27,0
/web/guest/dpto.-io	50	1739	1540	3047	371	4829	0,00%	2,0/sec	23,4
/web/guest/subdireccion	50	1857	1655	3065	241	5159	0,00%	1,9/sec	22,0
/web/guest/formacion	50	1824	1485	3669	359	6120	0,00%	1,9/sec	22,6
/web/guest/postgrado	50	1577	1179	2916	155	4266	0,00%	2,0/sec	23,5
/web/guest/wiki	50	11283	11744	21688	1360	24089	0,00%	1,4/sec	26,9
/web/guest/blogs	50	5368	5559	7255	1405	7622	0,00%	1,4/sec	6,9
/web/guest/comunidades	50	6677	5218	12766	2011	18204	0,00%	1,2/sec	38,4
/web/guest/liferay	50	1147	933	2259	205	4103	0,00%	1,3/sec	15,1
/web/guest/java	50	928	889	1558	88	2279	0,00%	1,3/sec	15,6
/web/guest/recursos	50	1024	823	1633	124	3966	0,00%	1,4/sec	15,7
/web/guest/herramientas	50	961	840	1730	95	2894	0,00%	1,4/sec	16,3
/web/guest/documentos	50	798	671	1437	95	2431	0,00%	1,4/sec	16,8
Total	850	4194	1646	11744	88	27993	0,00%	10,2/sec	150,6

Tabla 10. Informe Agregado. Portal CDAE.



**Gráfica 1. Portal CDAE.**

Para un total de 50 usuarios concurrentes por página y un total de 850: el tiempo máximo de respuesta fue 27s, asociado de un tiempo mínimo de 0.88s y un rendimiento de 10.2 peticiones por segundos.

El máximo de tiempo invertido por petición (columna Max) rondó los 27 segundos, con un mínimo de tiempo invertido por una petición (columna Min) de 0.88 segundos y una media de tiempo invertido por una petición (columna Media) de 4 segundos. La mediana de tiempo invertido por una petición, que significa que el 50% de las muestras tardaron menos del valor reflejado fue de 1.6 segundos con un porcentaje de respuestas con error igual al 0%. El rendimiento, número de peticiones procesadas en una unidad de tiempo, que puede expresarse segundos, minutos y horas fue de 10,2/sec.

### **3.4 Conclusiones del capítulo.**

En este capítulo se explicaron detalladamente los procesos llevados a cabo para la implementación de la solución propuesta. Fueron expuestos y explicados los métodos de pruebas que se utilizaron. Se muestran además el flujo de las tareas para cada una de las funcionalidades desarrolladas en el portal así como las pruebas realizadas a las aplicaciones obtenidas y al portal. Con la culminación de este capítulo se da por terminada la propuesta de solución de la aplicación a implementar.

## CONCLUSIONES

El presente trabajo se enfocó en el estudio y desarrollo de un portal web para la divulgación de la información relacionada con el CDAE y la gestión de la información relacionada con las evaluaciones periódicas así como los temas de tesis que se les brindan a los estudiantes vinculados al centro. Con este sistema los profesores podrán contar con una herramienta que les facilite el trabajo diario y les provea de información real y actualizada. Además posibilitará un ahorro considerable de tiempo, esfuerzo y coste en los que anteriormente se incurrían por falta de un proceso automatizado en la gestión de las evaluaciones.

Se desarrolló un marco teórico referencial relacionado con el uso de los portales como plataformas tecnológicas en las organizaciones, demostrando que Liferay Portal es la plataforma más práctica para ser implantada en el centro.

Se logró el diseño e implementación de un portal web con las siguientes ventajas:

- Ahorro en tiempo y recursos para las personas que utilicen la aplicación.
- Fácil mantenimiento, de modo que se le puedan adicionar nuevas aplicaciones.
- Un medio en el cual se puede compartir los conocimientos de la comunidad y para la divulgación de información relacionada con el centro.

Se validó la solución desarrollada mediante el uso de pruebas de aceptación, pruebas de carga y estrés, arrojando resultados satisfactorios.

## RECOMENDACIONES

Referente a los resultados obtenidos en la investigación efectuada durante la elaboración de este trabajo, y con la intención de asegurar la posterior ampliación, modificación y mejora del portal empresarial propuesto, se exponen a continuación algunas recomendaciones:

- Desarrollar aplicaciones compuestas que beneficien la optimización de procesos dentro del centro basadas en la plataforma que se ha propuesto en esta investigación.
- Integrarle al portal el servicio de correo de la universidad.

## BIBLIOGRAFÍA

1. Huertas, D.O.B., *Del Portal en la Comunicación Empresarial a la Web 2.0. La carrera por la empatía con el usuario*. 2008.
2. definicion.de, *Portal*.
3. Lara-Navarra, P., *Metamorfosis de los portales: del hiper-texto a la hiper-red*.
4. D., M.S. and G.A. F, *Prospectiva de los portales Corporativos*. 2007.
5. Palacios, E., *Aplicaciones ricas en Internet (RIA). Un enfoque de refactorización*.
6. Valdés, N.P., *Cuba y la tecnología de la información*. 2002.
7. Tonina, M., *¿Qué es un servidor de aplicaciones?*
8. Liferay, P.o.d., *Liferay*. 2011.
9. Junction, J., *JBOSS Portal vs Liferay Portal*.
10. González, Y.P., *Integración de Liferay con Openxava para el desarrollo de una plataforma para los proyectos del Sistema de Gestión del Convenio Integral de Cooperación*. 2011.
11. Grönroos, M., *Book of Vaadin*. 2010.
12. GARCÍA, J.L.R., *Primeros pasos para el mapeo entre modelo O.O y modelo Relacional usando Hibernate en JAVA*.
13. UCI, C.d.P., *Acerca de PostgreSQL*.
14. Hechevarria, B.H.C.y.D.F.C., *Diseño y evaluación de la Arquitectura de Software con la tecnología J2EE y el Framework Spring*. 2009.
15. Almenares, I.L.S., *Cómo realizar Pruebas de Carga y Estrés en JMeter*. 2008.
16. Silva, P.V.P., *Portal de la Comunidad Técnica Cubana de PostgreSQL V1.0. Implementación y Prueba*. 2011.
17. Liferay, I., *Liferay Developer's Guide*. 2011.
18. Montero, H.R. (2007) *Experiencia, evolución y Open Source*.
19. Sanchez, M.A.M., *Metodologías De Desarrollo De Software*. 2004.
20. Penadés, P.L.y.M.C., *Metodologías ágiles para el desarrollo de software: Extreme Programming (XP)*.
21. Penadés, P.L.y.M.C., *Métodologías ágiles para el desarrollo de software: Extreme Programming (XP)*.

22. Francisco J. Martín Fernández, Y.H.M., *Qué es la Arquitectura de la Información*. 2003: p. 1.
23. AGESIC, *Seguridad de un portal*.
24. Costa, D.C., *Introducción al diseño de bases de datos*.
25. González, A.H., *Un método para el diseño de la base de datos a partir del modelo orientado a objeto*. Vol. 7. 2004.
26. J. J. Gutiérrez, M.J.E., M. Mejías, J. Torres, *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA*.
27. Juan Pablo Cassinelli, R.S., Dayvis Malfará, Diego Cukerman, Fernando Cócaro, *Testing en eXtreme Programming*.
28. <http://www.corporacionsyben.com/>, *Pruebas de Rendimiento*.