



## **Facultad 5**

### **Centro de Consultoría y Desarrollo de Arquitecturas Empresariales**

**Portlet de autorización por roles de acciones de grano fino para  
aumentar la eficiencia en la gestión de la seguridad en los portlets  
desarrollados para el portal Liferay.**

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

**Autora: Susana Alba Silot**

**Tutor: Ing. Armando Masó Mosqueda**

**Co-Tutora: Ing. Yuriseidis María Reina Torres**

**“Año 57 de la Revolución” La Habana, Cuba, junio 2015**

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

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

---

Armando Masó Mosqueda  
**Tutor**

---

Yuriseidis María Reina Torres  
**Co-Tutora**

---

Susana Alba Silot  
**Autora**

**Datos de contacto**

**Tutor:** Amando Masó Mosqueda.

**Ciudadanía:** Cubana.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniero en Ciencias Informáticas.

**E-mail:** amaso@uci.cu.

**Co-Tutora:** Yuriseidis María Reina Torres.

**Ciudadanía:** Cubana.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniera en Ciencias Informáticas.

**E-mail:** ymreina@uci.cu.

**Dedicatoria**

A Dios y a mi familia, en especial a mi tía Vilma Silot, mi papá Braulio Manuel Alba, mi mamá Maritza Silot, mis hermanos Yasmani Alba, Yoenni Alba, John Alba y mi prima Yisel Pérez, por su apoyo incondicional y que no me dejaron desfallecer para así poder llevar a cabo la culminación de este trabajo.

## **Agradecimientos**

Primero y antes que nada, doy gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino aquellas personas que han sido mi soporte y compañía durante todo el período de estudio.

Agradezco a mi familia y a mi novio por el ánimo, apoyo y alegría que me brindaron, dándome fortaleza para seguir adelante.

Quiero agradecerles a mis tutores, por sus conocimientos invaluableles que me brindaron para llevar a cabo esta investigación, y sobre todo su gran paciencia para esperar a que este trabajo pudiera llegar a su fin, a los miembros del tribunal, por las valiosas contribuciones que hicieron al trabajo final y por el tiempo que dedicaron para revisarlo.

Agradezco a los excelentes profesores que me han impartido clases, así como a mis compañeros por todos los buenos y malos momentos que viví con ellos y a todas esas personas que me brindaron su colaboración, sus conocimientos, su ayuda incondicional y por sobre todo su amistad.

## Resumen

Lograr diferencias competitivas, obtener agilidad de negocio y desarrollar servicios mediante el uso de las nuevas tecnologías, constituyen necesidades de las empresas de hoy. Estas necesidades pueden erradicarse haciendo uso de diferentes herramientas, destacándose el uso de portales web (dirigidos a ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de servicios relacionados a un mismo tema. Incluye: enlaces, buscadores, foros, documentos, aplicaciones, compra electrónica, etc).

Liferay Portal es una de estas plataformas web, que tiene como objetivo ofrecer el máximo beneficio para sus clientes. Es un producto que presenta más de 60 aplicaciones portlets<sup>1</sup>, listas para desarrollar tareas básicas de un portal moderno a nivel de empresa.

En la presente investigación se argumenta el desarrollo de un portlet administrativo que permitirá a las entidades, administrar las autorizaciones basadas en roles de acciones de grano fino sobre los portlets desarrollados para el portal Liferay. El ciclo de desarrollo de la solución estuvo regido por la Metodología Unificada Ágil (AUP) adaptada a los procesos productivos de la Universidad de las Ciencias Informáticas (UCI). Las tecnologías y herramientas empleadas para la aplicación son libres (Open Source), con excepción de Visual Paradigm for UML.

**Palabras claves:** autorizaciones, acciones de grano fino, portlet administrativo.

---

<sup>1</sup> En la investigación se define portlet como componentes modulares o recurso.

## Índice de contenidos

Introducción.....	1
Capítulo #1. Fundamentación teórica .....	- 5 -
1.1    Introducción.....	- 5 -
1.2    Seguridad informática.....	- 5 -
1.3    Mecanismos básicos de seguridad.....	- 5 -
1.4    Política de seguridad .....	- 6 -
1.5    Modelos de seguridad.....	- 7 -
1.6    Gestión de roles y permisos.....	- 8 -
1.7    Liferay Portal .....	- 10 -
1.8    Portlets.....	- 10 -
1.9    Tecnologías, herramientas y metodología a utilizar. ....	- 11 -
1.7.1    Lenguaje de programación Java.....	- 11 -
1.7.2    Framework Vaadin .....	- 12 -
2.7.2    Framework JSF .....	- 12 -
1.7.3    Comparativa entre librerías de componentes visuales:[20] .....	- 13 -
2.7.3    Spring Framework .....	- 16 -
3.7.3    Framework Hibernate .....	- 18 -
4.7.3    Sistema Gestor de Bases de Datos .....	- 19 -
1.5.4.1    MySQL.....	- 19 -
1.5.4.2    PostgreSQL Server.....	- 20 -
5.7.3    Entorno de Desarrollo Integrado. Eclipse Luna.....	- 20 -
6.7.3    Servidor de Aplicaciones. Apache Tomcat. ....	- 21 -
7.7.3    Metodología de Desarrollo. Metodología Unificada Ágil (AUP). ....	- 21 -
8.7.3    Herramienta de Modelado UML. Visual Paradigm for UML. ....	- 23 -
1.6    Conclusiones parciales.....	- 24 -

---

<b>Capítulo #2. Características y diseño del sistema .....</b>	<b>- 25 -</b>
2.1 Introducción .....	- 25 -
2.2 Propuesta de solución.....	- 25 -
2.3 Modelado conceptual.....	- 25 -
2.3.1 Descripción del diagrama de clases del modelo conceptual.....	- 26 -
2.4 Modelado de requisitos del sistema .....	- 27 -
2.4.1 Requisitos funcionales (RF) .....	- 27 -
2.4.2 Requisitos no funcionales (RNF) .....	- 28 -
2.5 Historias de Usuario .....	- 28 -
2.5.1 Descripción de las Historias de Usuario .....	- 28 -
2.6 Estilo arquitectónico.....	- 30 -
2.6.1 Arquitectura de la aplicación .....	- 31 -
2.7 Patrón Modelo Vista Controlador.....	- 32 -
2.8 Patrones de diseño .....	- 33 -
2.8.1 Patrones GRASP .....	- 34 -
2.8.2 Patrones GOF .....	- 35 -
2.9 Modelo de diseño .....	- 35 -
2.9.1 Descripción de las clases del diseño .....	- 36 -
2.11 Conclusiones parciales.....	- 37 -
<b>Capítulo #3. Implementación y prueba del sistema .....</b>	<b>- 38 -</b>
3.1 Introducción .....	- 38 -
3.3 Estándar de código.....	- 38 -
3.4 Pruebas.....	- 39 -
3.4 Estrategia de pruebas.....	- 40 -
3.5 Casos de pruebas .....	- 46 -

3.6 Lista de chequeo[42].....	- 48 -
3.5.1 Resultados.....	- 49 -
3.6 Conclusiones parciales .....	- 50 -
Conclusiones generales .....	- 51 -
Recomendaciones.....	- 52 -
Bibliografía Referenciada.....	- 53 -
Anexos .....	- 56 -

## Índice de figuras

<i>Figura 1 Framework Spring. Generación de objetos.....</i>	<i>- 17 -</i>
<i>Figura 2 Framework Spring. Integración de objetos.....</i>	<i>- 18 -</i>
<i>Figura 3 Modelo Conceptual.....</i>	<i>- 26 -</i>
<i>Figura 4 Arquitectura en 3 Capas.....</i>	<i>- 31 -</i>
<i>Figura 5 Patrón Modelo Vista Controlador. ....</i>	<i>- 33 -</i>
<i>Figura 6 Diagrama de clases del diseño.....</i>	<i>- 35 -</i>

---

## Índice de tablas

<b>Tabla 1 Comparativa entre librerías de componentes visuales.....</b>	<b>- 16 -</b>
<b>Tabla 2 Descripción del diagrama de clases del modelo conceptual.....</b>	<b>- 26 -</b>
<b>Tabla 3 Requisitos funcionales.....</b>	<b>- 28 -</b>
<b>Tabla 4 Requisitos no funcionales.....</b>	<b>- 28 -</b>
<b>Tabla 5 HU Obtener listado de roles.....</b>	<b>- 29 -</b>
<b>Tabla 6 HU Obtener listado de portlets.....</b>	<b>- 30 -</b>
<b>Tabla 7 Descripción de las clases del diseño.....</b>	<b>- 36 -</b>
<b>Tabla 8 Estrategia de pruebas.....</b>	<b>- 46 -</b>
<b>Tabla 10 Lista de chequeo.....</b>	<b>- 49 -</b>

## Introducción

Las empresas de hoy necesitan poder interconectar los procesos, personas e información, aumentar los canales de interacción con el cliente y mejorar su experiencia y satisfacción. Además de lograr diferencias competitivas, agilidad en el negocio y desarrollar servicios a través de la utilización efectiva de nuevas tecnologías, que logren la integración de todos estos componentes.

En el mundo del desarrollo de software existe una herramienta que satisface muchas de estas necesidades, al proporcionar a las personas el acceso a información, aplicaciones y procesos de negocio; definiéndose como portal. Todo ello desde un único punto de entrada, que provee mecanismos de integración, inicio único de sesión, seguridad y contenido personalizado. Además de ofrecer preferencias a medida de los usuarios, gestión de contenidos, opciones de colaboración entre personas, acceso desde múltiples dispositivos, opciones de búsqueda y navegación entre aplicaciones.[1]

Liferay Portal es una de las plataformas web corporativa Open Source, más utilizada del mundo, proyectando en el 2013, más de cinco millones de descargas y una comunidad de más de 90.000 usuarios registrados.[2] Por quinto año consecutivo, la consultora independiente Gartner (empresa consultora y de investigación de las tecnologías de la información) ha posicionado a Liferay como líder en su Cuadrante Mágico.[3]

Las funcionalidades de éste podrían agruparse en dos conjuntos. Por un lado, la configuración estándar permite administrar las funcionalidades más básicas en cuanto a usuarios, grupos de usuarios, roles, organizaciones y comunidades. Por otra parte, provee un conjunto predefinido de herramientas denominado Liferay Collaboration Suite. Esta Suite contiene blogs, calendarios, mensajería instantánea, correo electrónico (para servidores ya existentes y que soporten *IMAP* (Protocolo de acceso a mensajes de internet), tablero de anuncios (foros) y wikis.[1] Compatible con todos los servidores de aplicaciones líderes y bases de datos, destacándose los Sistemas Gestores de Base de Datos PostgreSQL Server y MySQL, y sistemas operativos actuales como Linux, UNIX y Windows.

Además, cuenta con la posibilidad de incorporar plugins para extender las funcionalidades básicas. Entre ellas se encuentran los portlets (pequeñas aplicaciones web que se ejecutan en una parte de una página web), temas de diseño, distribución de elementos de páginas, y módulos

web en Java EE (Java Enterprise Edition, por sus siglas en inglés). Algunos de estos plugins ya vienen incorporados por defecto en la instalación estándar.[1]

Corporaciones internacionales como Smile (cuenta con nueve agencias en países como: Francia, España, Suiza, Ucrania y Marruecos), openTrends (tiene su sede en España) y Mexuz (tiene sede en Australia), entre otras, hacen uso de Liferay para desarrollar sitios web y portales que proporcionen soporte a sus negocios.

De igual forma, Cuba hace uso de esta herramienta con el objetivo de desarrollar aplicaciones de negocio, de forma rápida y sencilla. Proyectos como Sistema informativo del Ministerio de la Colaboración Económica en el Extranjero (SI-MINCEX), Gestión de operaciones del Centro de Inmunología Molecular (CIM) y el Sistema para la planificación de literaturas del Ministerio de Educación Superior (MES) (SOFTEXT) pertenecientes al Centro de Desarrollo de Arquitecturas Empresariales, de la Universidad de Ciencias Informáticas (UCI), son proyectos reales desarrollados con la plataforma Liferay. Esta plataforma posee potencia y flexibilidad al otorgar permisos de manera jerárquica y granular para gestionar los portlets. A pesar de estas características, no llega a un nivel donde se especifiquen permisos de grano fino sobre funcionalidades en estos componentes modulares, que posean mayor impacto en el portal, es el caso de registrar, editar, eliminar y visualizar algún tipo de información.

Algunos de los proyectos mencionados anteriormente gestionan las autorizaciones de acciones de grano fino de forma manual. Para el proyecto SI-MINCEX se desarrolló una clase de seguridad en la que se realizan consultas sobre la estructura del portal donde el usuario autenticado con un rol determinado tenga acceso a las funcionalidades del portlet. El proyecto SOFTEXT basó su seguridad en Spring Security, asegurando las direcciones URL (Uniform Resource Location, por sus siglas en inglés) de los módulos para el acceso a usuarios con roles específicos. En el proyecto para el CIM se hizo uso de una clase donde se implementaron un conjunto de funcionalidades que garantizan el acceso a los recursos. Estas gestiones manuales de seguridad traen consigo demoras en los tiempos de respuestas ante cambios en el negocio, dado que se tendría que actualizar el código fuente, compilarlo, probarlo y desplegarlo en el ámbito productivo. Dada la situación problemática antes planteada, se propone el **problema de investigación**:  
¿Cómo realizar la autorización basada en roles de acciones de grano fino para aumentar la eficiencia en la gestión de la seguridad de los portlets desarrollados para el portal Liferay?

**Objeto de estudio:**

La autorización basada en roles de acciones de grano fino.

**Objetivo General:**

Desarrollar un portlet que gestione las autorizaciones basadas en roles de acciones de grano fino para aumentar la eficiencia en la gestión de la seguridad en los portlets desarrollados para el portal Liferay.

**Campo de acción:**

La autorización basada en roles de acciones de grano fino en portlets desarrollados para el portal Liferay.

Para lograr el objetivo propuesto se presentan como **tareas investigativas:**

- Elaboración del marco conceptual para precisar los principales conceptos que se emplean durante la investigación.
- Elaboración del estado del arte sobre las tendencias actuales en la autorización basada en roles.
- Selección de la metodología para definir los métodos y técnicas necesarias que guiarán el desarrollo.
- Descripción de las herramientas, tecnologías y lenguajes a emplear para definir el ambiente de desarrollo.

**Idea a defender**

Con el desarrollo del portlet de autorización basada en roles de acciones de grano fino, se obtendrá mayor eficiencia en la gestión de la seguridad de los portlets desarrollados para el portal Liferay.

Para el desarrollo de las tareas de investigación se emplearon varios métodos:

**Métodos teóricos**

El **Análisis-Síntesis** para una mayor comprensión del funcionamiento del Liferay Portal, características, arquitectura que presenta y formular conclusiones a través de la síntesis de los conocimientos y resultados obtenidos.

**La Modelación** es considerada como uno de los métodos lógicos que consistirá en esquemas, diagramas o representaciones donde se refleje la estructura de relaciones y determinadas propiedades fundamentales del portlet.

**Métodos empíricos**

La **Observación Científica**, permite valorar los avances realizados en el estudio de las diferentes herramientas y tecnologías a usar.

El método de **Consulta de la Información** es utilizado para elaborar el marco teórico o el estado del arte de la investigación, permitiendo el conocimiento y el acceso a las referencias bibliográficas de los múltiples criterios que han sido citados para comprender mejor el problema de investigación planteado, y decidir si es posible científica y económicamente su ejecución.

### **Estructura de la tesis**

**Capítulo I.** Fundamentación Teórica. En este capítulo se estudian las tendencias actuales en el desarrollo de los portlets en el portal Liferay. Se define el lenguaje de programación, metodología, tecnologías y herramientas para la implementación de la aplicación que dará solución al problema planteado.

**Capítulo II.** Portlet Administrativo para Liferay como propuesta de solución que permitirá la manipulación de autorización de acciones de grano fino en portlets: tomándose como base la metodología AUP para guiar el proceso de desarrollo. Se da a conocer la propuesta del sistema y los diagramas para apoyar la comprensión del funcionamiento del mismo.

**Capítulo III.** Análisis y diseño del portlet Administrativo para el portal Liferay: Se muestra todo lo referente al flujo de implementación. Se diseñan, describen y realizan los casos de pruebas aplicados al sistema. También se valida el resultado mediante técnicas de validación.

## Capítulo #1. Fundamentación teórica

### 1.1 Introducción

En el capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se estudian las tendencias actuales; el uso de tecnologías, metodologías, arquitecturas y herramientas que brindan conocimiento y apoyo en el desarrollo del proceso de elaboración del producto final.

### 1.2 Seguridad informática

La seguridad informática es una disciplina que se relaciona a diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información un sistema informático y sus usuarios.

Técnicamente es imposible lograr un sistema informático ciento por ciento seguro, pero buenas medidas de seguridad evitan daños y problemas que pueden ocasionar intrusos.[4]

#### Seguridad con respecto a la naturaleza de la amenaza

Existen dos tipos de seguridad con respecto a la naturaleza de la amenaza: [4]

- Seguridad lógica: aplicaciones para seguridad, herramientas informáticas, etc.
- Seguridad física: mantenimiento eléctrico, anti-incendio, humedad, etc.

### 1.3 Mecanismos básicos de seguridad

#### Autenticación

Se define la autenticación como la verificación de la identidad del usuario, generalmente cuando entra en el sistema o la red, o accede a una base de datos. [5]

Es posible autenticarse de tres maneras:

- Por lo que se sabe (una contraseña).
- Por lo que se tiene (una tarjeta magnética).
- Por lo que la persona es (las huellas digitales).

#### Autorización

La autorización es el proceso por el cual se determina qué, cómo y cuándo, un usuario autenticado puede utilizar los recursos de la organización.[5]

Dependiendo del elemento la autorización puede hacerse por medio de la firma en un formulario o mediante una contraseña, pero siempre es necesario que dicha autorización quede registrada para ser controlada posteriormente. [5]

En el caso de los datos, la autorización debe asegurar la confidencialidad e integridad, ya sea dando o denegando el acceso en lectura, modificación, creación o borrado de los datos. [5]

Por otra parte, solo se debe dar autorización para acceder a un elemento a aquellos usuarios que lo necesiten para hacer su trabajo, y si no se le negará. Aunque también es posible dar autorizaciones transitorias o modificarlas a medida que las necesidades del usuario varíen. [5]

### **Administración**

Se define la administración como la que establece, mantiene y elimina las autorizaciones de los usuarios, los elementos y las relaciones usuarios-elementos del sistema. [5]

Los administradores son responsables de transformar las políticas de la organización y las autorizaciones otorgadas a un formato que pueda ser usado por el sistema. [5]

### **Auditoría y registro**

Se define la Auditoría como la continua vigilancia de los servicios en producción y para ello se obtiene información y se analiza. [5]

Este proceso permite a los administradores verificar que las técnicas de autenticación y autorización utilizadas se realizan según lo establecido y se cumplen los objetivos fijados por la organización. [5]

El registro: es un mecanismo por el cual cualquier intento de violar las reglas de seguridad establecidas queda almacenado en una base de eventos para luego analizarlo. [5]

### **Mantenimiento de la integridad**

El mantenimiento de la integridad de la información: es el conjunto de procedimientos establecidos para evitar o controlar que los archivos sufran cambios no autorizados y que la información enviada desde un punto llegue al destino inalterada. [5]

Dentro de las técnicas más utilizadas para mantener (o controlar) la integridad de los datos están: uso de antivirus, encriptación y funciones 'hash'. [5]

## **1.4 Política de seguridad**

Una política de seguridad es un plan de acción para afrontar riesgos de seguridad, e incluye las reglas, normas y prácticas que regulan cómo una institución gestiona y protege su

patrimonio. En la práctica, las políticas de seguridad fijan criterios a la hora de asignar permisos a los usuarios relativos a la capacidad de acceso a los diferentes elementos del sistema.[6]

### 1.5 Modelos de seguridad

La seguridad de una organización puede estar centrada en diferentes objetivos, como Confidencialidad, Integridad y Disponibilidad, dando lugar a diferentes modelos de seguridad. Respecto a las reglas de control de accesos en que se materializa una política de seguridad, existen varios modelos de control de accesos:[6]

**Modelo de control de acceso discrecional (DAC):** las reglas de acceso a un elemento son establecidas por el propietario del objeto, que decide quién tiene acceso al elemento y con qué permisos, por lo que el acceso se basa en la identidad del usuario. Este modelo utiliza el concepto de propietario del objeto y asignación de permisos. Se puede implementar mediante sistemas de control de accesos basados en listas de control de acceso (ACL) o basados en capacidades. Este es un método de control de acceso poco restrictivo, ya que los propios usuarios establecen los permisos según su criterio.

#### Granularidad[7]

- Grano grueso: Significa que se asignan privilegios masivamente. Proporciona una rebaja en la carga de trabajo.
- Grano fino: Significa que se asignan privilegios de manera individual, a usuarios o archivos concretos.

El sistema de control de acceso debería proporcionar ambos.

**Modelo de control de acceso obligatorio o no-discrecional (MAC):** los permisos de acceso del usuario los asigna el administrador del sistema, por lo que el propietario del elemento no puede realizar ninguna acción contraria a la política del administrador. Generalmente se basa en un mecanismo de etiquetas, de tal manera que el sistema asocia una etiqueta de sensibilidad a cada elemento que se utiliza en las decisiones de control de acceso. Este es el método de control de acceso más restrictivo, ya que las reglas de acceso están centralizadas en el administrador del sistema.

**Modelo de control de acceso basado en roles (RBAC):** el acceso está basado en los permisos del rol que el sistema ha asignado al usuario. Los elementos precisan que el usuario tenga un rol determinado para poder acceder a los mismos.

**Modelo de control de acceso basado en atributos (ABAC):** el acceso no está basado en los permisos del usuario autenticado, sino en los atributos del usuario, por ejemplo, tener más de 18 años. Cualquier usuario con ese atributo tiene acceso, lo que permite incluso el acceso anónimo, ya que la identificación y autenticación no es estrictamente necesaria.

### 1.6 Gestión de roles y permisos

El Control de Acceso Basado en Roles (RBAC) es un enfoque para restringir acceso a un sistema a usuarios autorizados. Es un nuevo enfoque alternativo al acceso de control mandatorio (MAC) y acceso de control discrecional (DAC). RBAC es referido algunas veces como seguridad basada en roles.[8]

RBAC es una tecnología de control de acceso flexible y de política neutral suficientemente poderosa para simular DAC y MAC. De manera inversa, MAC puede simular RBAC si el gráfico de roles está restringido a un árbol en vez de un conjunto parcialmente ordenado.[8]

Anterior al desarrollo de RBAC, MAC y DAC eran considerados como los únicos modelos conocidos para el control de acceso: si un modelo no era MAC, se consideraba que era DAC, y viceversa. Investigaciones a finales de los 90 demostraron que RBAC no cae en ninguna de las dos categorías.[8]

Dentro de una organización son creados roles para varias funciones de trabajo. Los permisos para realizar ciertas operaciones son asignados a roles específicos. A miembros del personal (u otros usuarios del sistema) se les asigna roles particulares, y a través de la asignación de esos roles adquieren los permisos para realizar funciones de sistemas particulares. A diferencia de control de acceso basado en contexto (CBAC), RBAC no revisa el contexto del mensaje (tal como la fuente de la conexión).[8]

El modelo RBAC en Oracle Solaris introduce los siguientes elementos:[9]

- Autorización: un permiso para que un usuario o un rol realice una clase de acciones que requieren derechos adicionales. Por ejemplo, la política de seguridad en la instalación otorga a los usuarios comunes la autorización `solaris.device.cdrw`. Esta autorización permite a los usuarios leer y escribir en un dispositivo de CD-ROM.
- Privilegio: un derecho perfectamente definido que se puede otorgar a un comando, un usuario, un rol o un sistema. Los privilegios permiten que un proceso se realice correctamente. Por ejemplo, el privilegio `proc_exec` permite a un proceso llamar `execve()`. Los usuarios comunes tienen privilegios básicos.

- Atributos de seguridad: un atributo que permite a un proceso efectuar una operación. En un entorno UNIX típico, un atributo de seguridad permite a un proceso efectuar una operación que, de lo contrario, está prohibida para los usuarios comunes. Por ejemplo, los programas `setuid` y `setgid` tienen atributos de seguridad. En el modelo RBAC, las operaciones que los usuarios comunes realizan pueden requerir atributos de seguridad. Además de los programas `setuid` y `setgid`, las autorizaciones y los privilegios también son atributos de seguridad en el modelo RBAC. Por ejemplo, un usuario con la autorización `solaris.device.allocate` puede asignar un dispositivo para uso exclusivo. Un proceso con el privilegio `sys_time` puede manipular la hora del sistema.
- Aplicación con privilegios: una aplicación o un comando que puede anular los controles del sistema mediante la comprobación de atributos de seguridad. En un entorno UNIX típico y en el modelo RBAC, los programas que usan `setuid` y `setgid` son aplicaciones con privilegios. En el modelo RBAC, los programas que necesitan privilegios o autorizaciones para ejecutarse correctamente también son aplicaciones con privilegios.
- Perfil de derechos: una recopilación de capacidades administrativas que se pueden asignar a un rol o a un usuario. Un perfil de derechos puede constar de autorizaciones, comandos con atributos de seguridad y otros perfiles de derechos. Los perfiles de derechos ofrecen una forma práctica de agrupar los atributos de seguridad.
- Rol: una identidad especial para ejecutar aplicaciones con privilegios. Sólo los usuarios asignados pueden asumir la identidad especial. En un sistema que se ejecuta por roles, el súper usuario resulta innecesario. Las capacidades de súper usuario se distribuyen en roles diferentes. Por ejemplo, en un sistema de dos roles, las tareas de seguridad serían gestionadas por un rol de seguridad. El segundo rol se ocuparía de las tareas de administración del sistema que no están relacionadas con la seguridad. Los roles pueden ser más específicos. Por ejemplo, un sistema podría incluir roles administrativos independientes para gestionar la estructura criptográfica, las impresoras, la hora del sistema, los sistemas de archivos y la auditoría.

Luego de realizar un estudio de los diferentes modelos de seguridad e investigar el cómo se maneja la gestión de los permisos en los diferentes proyectos (Sistema informativo del MINCEX (Ministerio de la Colaboración Económica en el Extranjero), Proyecto piloto del CIM (Centro de Inmunología Molecular) y Sistema para la planificación de literaturas del MES

(Ministerio de Educación Superior) SOFTEXT) pertenecientes al CDAE. Se opta por realizar un control de acceso híbrido (RBAC + ABAC) que gestione a nivel de interfaz las autorizaciones.

Se plantea que un sistema que involucra a todo o parte del personal de una empresa o institución, requiere de un control estricto para acceder a la información y el grado de permisos que se quiere asignar a cada empleado.

Al definir la organización como una estructura jerárquica, se puede precisar qué parte de la jerarquía puede visualizar un empleado, pero que no pueda ascender por la misma, más allá de lo que se le ha autorizado. Por tanto, diferentes roles permiten asignar a cada usuario lo que pueden hacer con la información, sólo consultarla, eliminarla o modificarla, visualizar los datos de carácter personal o que éstos permanezcan ocultos.

### **1.7 Liferay Portal**

Liferay es un portal web Open Source para la creación y gestión de escritorios personalizados que cuenta con una elevada escalabilidad y robustez para el manejo de las aplicaciones empresariales de todos los sectores. [10]

Liferay está basado en los principios de la Arquitectura Orientada a Servicios (SOA), ofrece escalabilidad, fiabilidad y alto rendimiento, tanto en la nube como "on premise" (software instalado y operado desde un servidor local ).[3] Soporta millones de usuarios online. Diseñado para que el contenido se muestre perfectamente en cualquier tamaño de pantalla. Se encuentra en más de veinte idiomas y proporciona una funcionalidad completa para la traducción de cada elemento del portal. Permite metodologías de escalado horizontal y vertical. Implementa las 10 prácticas principales de seguridad recomendadas por The Open Web Application Security Project (OWASP), entre otras características destacables.[11]

Hernández expresa que, "la adecuación de este portal a los estándares J2EE hace posible implementar esta tecnología en multitud de contextos sin que existan restricciones a nivel de sistema operativo, servidor de aplicaciones o base de datos." [12]

### **1.8 Portlets**

Los portlets son componentes modulares de interfaz de usuario gestionada y visualizada en un portal web. Producen fragmentos de código de marcado (html, xhtml) que se agregan en una página de un portal.[10] Además, una página de un portal se visualiza como una colección

de ventanas de portlet que no se solapan, donde cada una de estas muestra un portlet. Por lo tanto un portlet (o colección de portlets) se asemeja a una aplicación web que está hospedada en un portal.

Los portlets son módulos Web reutilizables que se ejecutan en un servidor del portal y proporcionan acceso al contenido, a las aplicaciones y a otros recursos basados en la web. Además, se ensamblan en una página más grande, con varias instancias del mismo portlet que muestran diferentes datos para cada usuario.[13]

Características principales:

- Los portlets son manejados por un contenedor especializado, que controla su ciclo de vida.
- Los portlets generan contenido dinámico e interactúan con el cliente web mediante el uso del paradigma request/response (solicitud/respuesta).
- Los portlets son únicamente generados como parte de una página web y no como documentos completos, no están asociados directamente a una URL y no pueden generar contenido arbitrario, es por estas razones que son diferentes a los servlets.
- Los portlets proporcionan un objeto Portlet Preferences para almacenar las preferencias del usuario.

## **1.9 Tecnologías, herramientas y metodología a utilizar.**

### **1.7.1 Lenguaje de programación Java**

Como el código compilado de Java (conocido como bytecode) es interpretado, un programa compilado puede ser utilizado por cualquier computadora que tenga instalado el intérprete de Java. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que instalar un intérprete para cada plataforma. De esa manera logra ser un lenguaje que no depende de una arquitectura computacional disponible. Además, no se requiere que se compilen todas las clases de un programa para que este funcione.[14]

Debido a la Máquina Virtual de Java (JVM) las aplicaciones desarrolladas en Java funcionan en Linux, Windows, Mac OS, y en cualquier sistema operativo para el cual exista una JVM, garantizando que no se dependa de un único sistema operativo para el funcionamiento de nuestra solución.[14]

Java es un lenguaje de programación orientado a objetos, de arquitectura neutral, por lo que es altamente portable, segura, de multihilos y dinámica.[15]

Dado que Liferay es un portal de gestión de contenidos de código abierto escrito en Java, se escoge para el desarrollo del proyecto el lenguaje de programación Java para continuar la misma estructura.

### **1.7.2 Framework Vaadin**

Vaadin es un framework para el desarrollo de aplicaciones web Open Source modernas o también llamadas Rich Internet Applications (RIA), que posee una librería con numerosos componentes de usuario, que construye aplicaciones web profesionales similares a los interfaces de una aplicación de escritorio. Este framework se ejecuta del lado del servidor, lo que significa que la lógica y la mayor carga del trabajo recaen en el servidor. [16]

Vaadin utiliza tecnología Ajax del lado del cliente cuya base es Google Web Toolkit (GWT) para renderizar la interfaz de usuario en el navegador. Las aplicaciones GWT se escriben en Java y son automáticamente traducidas a JavaScript y HTML. GWT es muy cómodo para el desarrollador porque le mantiene al margen de estas acciones de presentación y sin la necesidad de instalar ningún plugin para su presentación.[16]

### **2.7.2 Framework JSF**

Java Server Faces (JSF) es un framework de desarrollo basado en el patrón Modelo Vista Controlador (MVC). [17]. Según Alejandro Pérez García: “Al igual que el framework Struts, JSF pretende normalizar y estandarizar el desarrollo de aplicaciones web. Hay que tener en cuenta que JSF es posterior a Struts, y por lo tanto se ha nutrido de la experiencia de éste, mejorando algunas de sus deficiencias. De hecho el creador de Struts (Craig R. McClanahan) también es líder de la especificación del framework.”

JSF es un marco de trabajo para crear aplicaciones java J2EE. Permite desarrollar rápidamente aplicaciones de negocio dinámicas en las que toda la lógica de negocio se implementa en java, o es llamada desde java, creando páginas para las vistas muy sencillas (salvo que se introduzca mucha maquetación HTML o JavaScript).[18]. Simplifica el desarrollo de interfaces gráficas de usuario (GUI) en aplicaciones Java EE.

Esta tecnología surgió como una solución a la separación entre la presentación y el comportamiento en una aplicación Web, de tal forma que las actividades de los autores de las páginas puedan separarse de las actividades de los desarrolladores de la lógica.

JSF 2 se construye bajo la experiencia práctica de librerías de componentes visuales como ICEfaces, RichFaces, PrimerFaces, MyFaces. Consiguiendo de esta manera simplificar el desarrollo como una opción de futuro.

En su mayoría, las aplicaciones JSF son como cualquier otra aplicación web Java. Se ejecutan en un contenedor de servlets (programa Java que se ejecuta dentro de un servidor web, que reciben y atienden las solicitudes de los clientes web) y, típicamente, contienen:[19]

- Componentes Java Beans (llamados objetos del modelo en tecnología JSF) conteniendo datos y funcionalidades específicas de la aplicación.
- Oyentes de Eventos.
- Páginas, (principalmente páginas JSP).
- Clases de utilidad del lado del servidor, como beans para acceder a las bases de datos.

Además de estos ítems, una aplicación JSF también tiene:

- Una librería de etiquetas personalizadas para dibujar componentes UI (Interfaz de Usuario) en una página.
- Una librería de etiquetas personalizadas para representar manejadores de eventos, validadores y otras acciones.
- Componentes UI representados como objetos con estado en el servidor.

Características de JSF 2.0:

- Soporte para AJAX.
- Ámbito de los BackBeans.
- Declaración de los BackBeans e Inyección de Dependencias.
- Navegación.

Realizado el estudio de los frameworks de interfaz visual se escoge JSF para el desarrollo de la aplicación, por permitir el uso de diferentes tecnologías como JSP, xhtml, etc. Además proporciona una arquitectura que permite gestionar y procesar el estado de los componentes, validar la entrada del usuario y gestionar eventos.

### 1.7.3 Comparativa entre librerías de componentes visuales:[20]

Características:	PrimeFaces	ICEFaces	RichFaces

<b>Soporte de Ajax</b>	Es transparente para el desarrollador, aunque para activarlo deben utilizarse atributos específicos para lanzar un método del servidor y para indicar los componentes a actualizar.	Es transparente para el desarrollador, lo implementa de forma nativa en todos los componentes mediante la propiedad <code>partialSubmit</code> .	Tenemos que hacer uso de <code>Ajax4JSF</code> , que no es tan transparente para el desarrollador, puesto que, además de introducir los componentes de <code>RichFaces</code> , tenemos que añadir componentes no visuales de la librería <code>Ajax4JSF</code> .
<b>Librerías en las que se basan</b>	Utiliza el soporte de <code>jQuery</code> y <code>jQuery UI</code> para los efectos visuales.	Usa el soporte de <code>prototypejs</code> , aunque la parte de Ajax la ha rescrito y para los efectos visuales utilizan <code>script.aculo.us</code> .	Usa el soporte de <code>prototypejs</code> y <code>script.aculo.us</code> , aunque soporta también <code>jQuery</code> .
<b>Personalización de la interfaz de usuario</b>	Incorpora el concepto de skin, utilizando <code>ThemeRoller</code> , y dispone de 26 temas prediseñados.	Incorpora el concepto de skin y distribuye 3 temas.	Incorpora el concepto de skins y distribuye 12 temas, aunque se pueden encontrar más en el repositorio de <code>SNAPSHOTs</code> .
<b>Número de componentes</b>	Tiene más de 90 componentes OpenSource,	Tiene 79 componentes en la versión básica, a los	Tiene 212 componentes entre los propios de <code>RichFaces</code> y los de

	<p>algunos muy avanzados como el HTML Editor. Además dispone de un kit para crear interfaces web para teléfonos móviles.</p>	<p>que hay que sumar 32 de la versión empresarial, esta última es de pago. La percepción es que están invirtiendo esfuerzos en mejorar la versión empresarial y, como es lógico, esperan obtener beneficio económico por ello.</p>	<p>Ajax4JSF. Con RichFaces todos los componentes son OpenSource y podemos usar un Pick List sin contratar nada, sin embargo, con ICEfaces si queremos un Dual List o pagamos o lo implementamos nosotros.</p>
<b>Licencia</b>	<p>Apache License V2</p>	<p>MPL 1.1, que cubre la LGPL V 2.1. Si bien disponen de una versión empresarial con licencia comercial.</p>	<p>LGPL V 2.1 en su totalidad.</p>
<b>Relevancia</b>	<p>Ha sido una de las primeras librerías capaces de integrarse con JSF 2 y viene pisando fuerte debido a la diversidad y calidad de sus componentes. Puede utilizarse junto a RichFaces,</p>	<p>Ha sustituido a Woodstock como librería de componentes de referencia de Sun para el desarrollo de aplicaciones RIA. Se distribuye, por defecto, con NetBeans.</p>	<p>Es la librería de componentes visuales de Jboss, se integra, por defecto con Jboss Seam, aunque éste también soporta ICEfaces.</p>

	pero no es compatible con ICEfaces.		
--	-------------------------------------	--	--

**Tabla 1 Comparativa entre librerías de componentes visuales.**

Podría decirse que la librería de código abierto para JSF PrimeFaces, cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Controla qué elementos de la página actual se actualizarán y cuáles no, debido al soporte de Ajax con despliegue parcial que posee y permite la integración con otros componentes como por ejemplo: RichFaces.

Por lo antes expuesto y por ser estable y poseer una gran calidad de componentes se escoge la tecnología PrimeFaces, para el desarrollo de la interfaz visual del portlet administrativo.

### 2.7.3 Spring Framework

Spring es un framework liviano y no intrusivo: generalmente los objetos programados no tienen dependencias en clases específicas de Spring. Sus características principales son inyección de dependencias y programación orientada a aspectos. [21]

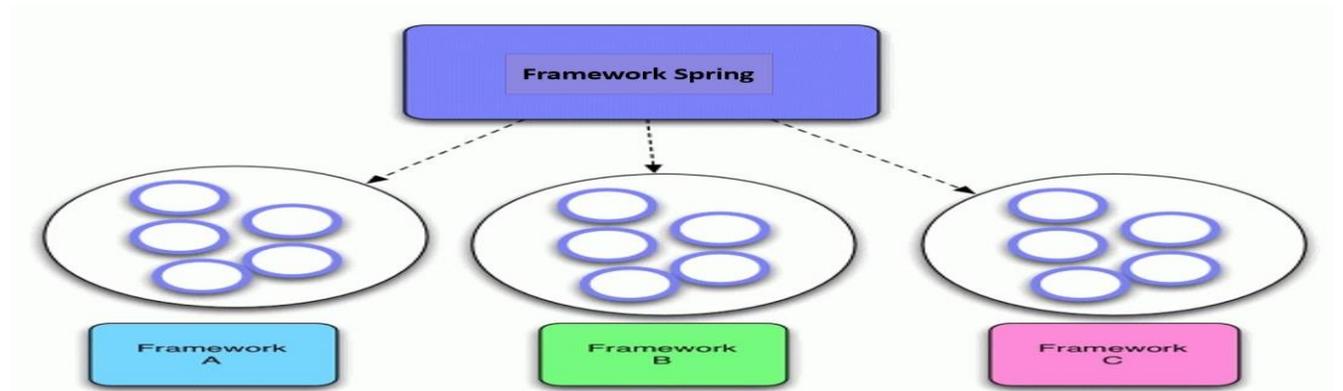
Spring es un framework ligero formado por un conjunto de piezas de diversa índole: Modelo Vista Controlador (MVC), seguridad, transaccionalidad, Programación Orientada a Aspecto (AOP) , integración con Web Services, Hibernate, etc; sin embargo, su componente principal, consiste en un contenedor de beans que permite, mediante Inversión de Control (IoC), acoplar las diversas partes de una aplicación.[22]

Este framework está diseñado para no ser intrusivo, esto significa que no es necesario que el sistema extienda o implemente alguna clase o interface de Spring, por lo que el código de lógica quedará libre y completamente reutilizable para un proyecto que no lo utilice, o por si se debe quitar de una aplicación que ya lo esté usando. Gracias a esto es posible usar un POJO o un objeto Java para realizar tareas que antes solo podían ejecutarse con Enterprise Java Beans (EJBs, una de las API que forman parte del estándar de construcción de aplicaciones empresariales Java Enterprise Edition). Sin embargo la utilidad de este framework, no es solo para el desarrollo de aplicaciones web, o no solo en el servidor, sino que permite que cualquier aplicación Java pueda beneficiarse de su uso.

Además, al usarlo de la forma correcta la aplicación quedará dividida en capas bien delimitadas, y con buenas prácticas de programación.

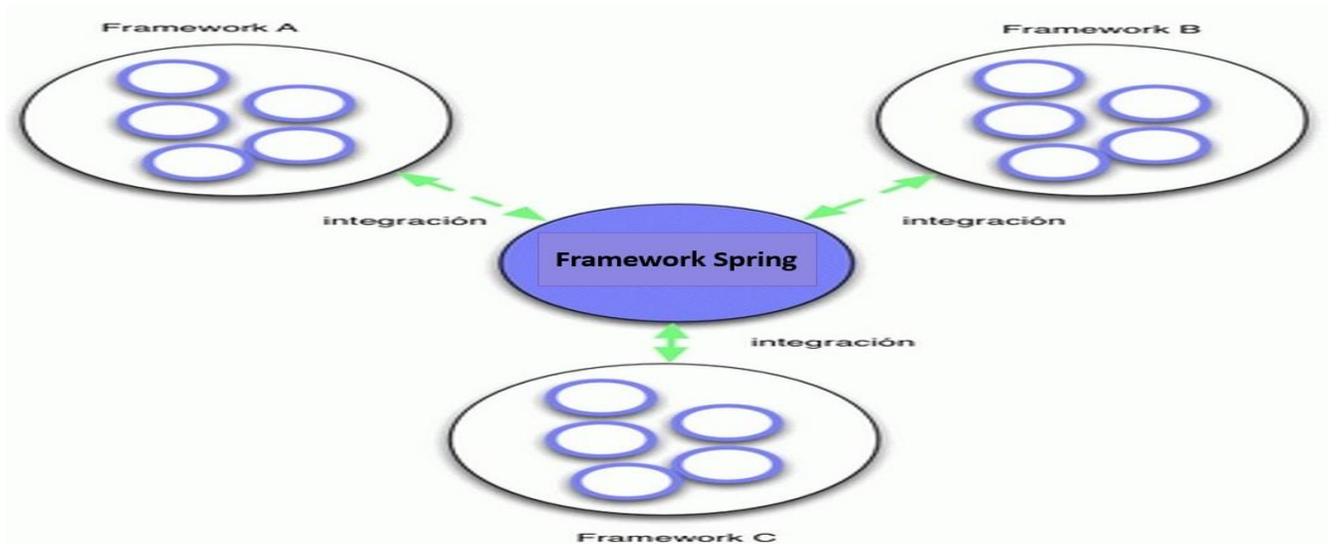
El núcleo de Spring está basado en un principio o patrón de diseño llamado Inversión de Control (IoC). Las aplicaciones que usan este principio se basan en su configuración (que en este caso puede ser en archivos XML o con anotaciones como en Hibernate) para describir las dependencias entre sus componentes. En este caso “inversión” significa que la aplicación no controla su estructura, permitiendo que sea el framework de IoC (Spring) quien lo haga.

Se puede decir que Spring cambia las responsabilidades y en vez de que el propio desarrollador sea el encargado de generar los objetos de cada uno de los frameworks, es éste basándose en ficheros XML o anotaciones, el encargado de construir todos los objetos que la aplicación va a utilizar.



**Figura 1 Framework Spring. Generación de objetos.**

De esta manera, Spring es también el responsable de asegurar que estos objetos se integren de la forma correcta.



**Figura 2 Framework Spring. Integración de objetos.**

### 3.7.3 Framework Hibernate

Hibernate permite desarrollar clases persistentes a partir de clases comunes, incluyendo asociación, herencia, polimorfismo, composición y colecciones de objetos. El lenguaje de consultas de Hibernate HQL (Hibernate Query Language), diseñado como una mínima extensión orientada a objetos de SQL, proporciona un puente elegante entre los mundos objetual y relacional. Hibernate también permite expresar consultas utilizando SQL nativo o consultas basadas en criterios. Soporta todos los sistemas gestores de bases de datos SQL y se integra de manera elegante y sin restricciones con los más populares servidores de aplicaciones J2EE y contenedores web, y por supuesto también puede utilizarse en aplicaciones standalone (aplicación que puede ser usada en cualquier computadora sin necesidad de ser instalada o que no requiere necesariamente una conexión a la red para funcionar).[10]

Hibernate es un Framework que agiliza la relación entre la aplicación y la base de datos. Para poder aprender a utilizarlo es necesario contar con los conocimientos básicos de base de datos y SQL así como manejar el lenguaje Java.[23]

Se considera entonces, que las características principales de este framework son las siguientes:

- Hibernate puede operar proporcionando persistencia de una manera transparente para el desarrollador.

- Soporta el paradigma de orientación a objetos de una manera natural: herencia, polimorfismo, composición y el framework de colecciones de Java.
- Permite una gran variedad de mapeos para colecciones y objetos dependientes.
- No es necesaria la generación de código ni el procesamiento del bytecode en el proceso de compilación.
- Este lenguaje proporciona una independencia del SQL de cada base de datos, tanto para el almacenamiento de objetos como para su recuperación.
- Soporte para transacciones de aplicación.
- Soporta los diversos tipos de generación de identificadores que proporcionan los sistemas gestores de bases de datos, así como generación independiente de la base de datos, incluyendo identificadores asignados por la aplicación o claves compuestas.
- Escalabilidad extrema.

### 4.7.3 Sistema Gestor de Bases de Datos

Un sistema de gestión de bases de datos (SGBD) es un sistema de software que permite el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Brindan funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

#### 1.5.4.1 MySQL

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple o multihilo, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es patrocinado por una empresa privada. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle

Corporation desde abril de 2009 desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad.

#### **1.5.4.2 PostgreSQL Server**

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales de código abierto y orientado a objetos. Brinda un control de concurrencia multi-versión, que permite trabajar con grandes volúmenes de datos. Funciona en todos los sistemas operativos Linux, UNIX y Windows.

El código fuente se encuentra disponible para todos sin costo alguno. Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir con fines privativo, comercial, así como académico. Por estas características es considerado como uno de los gestores de bases de datos más avanzado del mundo.

Ambos Gestores de Bases de Datos son potentes y pueden utilizarse para el desarrollo del proyecto, en este caso se trabajará con PostgreSQL Server por ser muy usado en entornos de software libre y puede funcionar en múltiples plataformas.

#### **5.7.3 Entorno de Desarrollo Integrado. Eclipse Luna.**

Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plugins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. Es un IDE genérico, aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el plugin JDT que viene incluido en la distribución estándar del IDE.[24]

Eclipse es una plataforma de desarrollo integrado, de código abierto y multiplataforma, potente y completa de programación, desarrollo y compilación de elementos variados como: sitios web, programas en C++ o aplicaciones Java. Es usado principalmente en la construcción de Aplicaciones de Cliente Enriquecido, opuesto a las aplicaciones Cliente-Liviano, basadas en navegadores. Posee una atractiva interfaz que lo hace fácil y agradable de usar.

Permite la instalación de plugins como es el caso del plugging Liferay IDE, brindando las opciones de desarrollar aplicaciones portlets y temas de interfaz visual, por lo que se escogió para el desarrollo de la solución propuesta.

### **6.7.3 Servidor de Aplicaciones. Apache Tomcat.**

Apache Tomcat es un servidor web multiplataforma que funciona como contenedor de servlets y se desarrolla bajo el proyecto Jakarta perteneciente a la Apache Software Foundation. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la licencia Apache 2.0 y que implementa las especificaciones de los servlets y de Java Server Pages (JSP).

Dado que Apache Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Además, la autora lo escoge como servidor web, por poseer ayuda y abundante documentación, cuenta con una comunidad de usuarios, es libre y es uno de los más usados en el CDAE.

### **7.7.3 Metodología de Desarrollo. Metodología Unificada Ágil (AUP).**

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información.[25]

Para guiar el proceso de desarrollo de la solución propuesta se escogió la Metodología Unificada Ágil (AUP), haciendo una variación de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

La metodología unificada ágil (AUP) es una forma simplificada del Proceso Racional Unificado (RUP, Rational Unified Process) desarrollada por Scott Ambler. Describe un enfoque simple del desarrollo del software usando técnicas y conceptos ágiles. Algunas técnicas usadas por AUP incluyen el desarrollo orientado a pruebas, modelado y gestión de cambios ágiles y refactorización de base de datos para mejorar la productividad.[26]

AUP establece cuatro fases: (Inicio, Elaboración, Construcción, Transición), la UCI decide para el ciclo de vida de los proyectos mantener la fase de Inicio, en la cual se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. Se unifican las restantes 3 fases de AUP en una sola, a la que se denomina

Ejecución que sería aquella fase en la que se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software, y se agrega una fase de cierre. En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.[27]

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 8 disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación y la disciplina Despliegue se considera opcional. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto).[27]

Ventajas:[26]

- El personal sabe lo que está haciendo: no obliga a conocer detalles.
- Simplicidad: apuntes concisos.
- Agilidad: procesos simplificados del RUP.
- Centrarse en actividades de alto valor: esenciales para el desarrollo.
- Herramientas independientes: a disposición del usuario.
- Fácil adaptación de este producto: de fácil acomodo (HTML).

Desventajas:[26]

- El AUP es un producto muy pesado en relación al RUP.
- Como es un proceso simplificado, muchos desarrolladores eligen trabajar con RUP, por tener a disposición más detalles en el proceso.

### 8.7.3 Herramienta de Modelado UML. Visual Paradigm for UML.

Visual Paradigm es una herramienta para UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.[28]

Visual Paradigm for UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (Visual Paradigm) Teniendo en cuenta sus características y los beneficios que brinda para la construcción de software, especialmente referente al modelado, se decidió utilizar Visual Paradigm for UML para el modelado de la extensión. Además de ello, se tuvo en cuenta que esta constituye la herramienta que utiliza la Universidad, y dentro de ella el centro DATEC para el desarrollo de software.[29]

Visual Paradigm for UML ofrece una variedad de funcionalidades muy ricas y potentes, como son:

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de usos.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidad de ingeniería directa e inversa.
- El modelado y el código permanecen sincronizado en todo el ciclo de desarrollo.
- Presenta una licencia gratuita y comercial.
- Se encuentra en varios idiomas.
- Soporta aplicaciones web.
- Potente generador de informes en formato PDF/HTML.
- Facilidad de instalación y de actualización.
- Compatibilidad entre ediciones.
- Interoperabilidad con modelos UML2 (metamodelos UML2 para plataforma Eclipse) a través de XML.

- Soporte ORM- Generación de Objetos Java desde bases de datos.
- Generación de bases de datos – Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Integración con Visio.
- Editor de figuras.
- Ambiente visual superior de modelado.
- Otros.

### 1.6 Conclusiones parciales

- Se logró realizar un análisis de los conceptos referentes a seguridad informática, política de seguridad, mecanismos básicos de seguridad y los diferentes modelos de seguridad existentes, así como las herramientas, tecnologías y la metodología de guía para el desarrollo de la solución.
- Se decide usar los frameworks Spring 4.1, Hibernate 4.3.6, JSF 2.1.3, se recomienda esta combinación dado que Spring Source sugiere PrimeFaces para ser utilizado en aplicaciones JSF -Spring, así que se escoge la librería de componentes visuales PrimeFaces 5.1 por ser estable y poseer una gran calidad de componentes, por lo que se toma como marco de interfaz de usuario, Spring para la capa de negocios e Hibernate como acceso a datos. Lo principal de esta combinación es, que hay configuraciones muy mínimas a través de XML y por anotaciones, dando más flexibilidad a los desarrolladores sólo para concentrarse en la interfaz de usuario y negocio.
- Se utiliza como Gestor de Bases de Datos PostgreSQL Server 9.2.4, el lenguaje de programación Java y la metodología AUP debido a su adaptabilidad a cualquier tipo de proyecto, es una forma simplificada de RUP y se preocupa por la gestión de riesgos.
- Las herramientas que serán usadas para complementar el trabajo serán: Visual Paradigm for UML 8.0 para el modelado UML, el entorno de desarrollo integrado Eclipse Luna Service Release 1 (4.4.1), como servidor web el Apache Tomcat 7.0.23 y el plugin de Liferay 6.1.1.

## Capítulo #2. Características y diseño del sistema

### 2.1 Introducción

En el presente capítulo se dará a conocer la propuesta del sistema y los diagramas para apoyar la comprensión del funcionamiento del mismo. Tomándose como base la metodología AUP para guiar el proceso de desarrollo. También se definirá la arquitectura que se usará para la implementación del portlet de administración de permisos de acciones de grano fino en estos mismos recursos.

### 2.2 Propuesta de solución

Liferay Portal actualmente no cuenta con un módulo que permita la gestión basada en roles de acciones de grano fino realizadas a las funcionalidades existentes en los portlets, obligando de esta forma a delegar estos permisos de forma manual.

Para resolver este problema se propone desarrollar un portlet que permita las autorizaciones por roles de acciones de grano fino que podrán realizarse sobre estos recursos para aumentar la eficiencia en la gestión de la seguridad.

Permitiendo obtener los portlets desarrollados, almacenarlos y asignarle las acciones de grano fino que se puedan desarrollar en éste, en caso de querer agregarle una acción que no esté registrada, el sistema permitirá registrarla y luego podrá agregarse a la lista de acciones sobre estos componentes modulares, además, se podrán obtener los roles existentes en el Liferay y por último gestionar los permisos que estos tendrán sobre los portlets.

A partir de las restricciones de negocio existentes, con el desarrollo de la propuesta de solución planteada, los administradores de sistemas tendrán la posibilidad de establecer niveles de acceso a la información de forma dinámica, sin necesidad de realizar modificaciones en el código de las aplicaciones.

### 2.3 Modelado conceptual

Un modelo conceptual es un conjunto de conceptos y de reglas destinados a representar de forma global los aspectos lógicos de los diferentes tipos de elementos existentes en la realidad que está siendo analizada. Es una representación figurada de una experiencia empírica, que tiene como objetivo ayudar a comprender la realidad.[30]

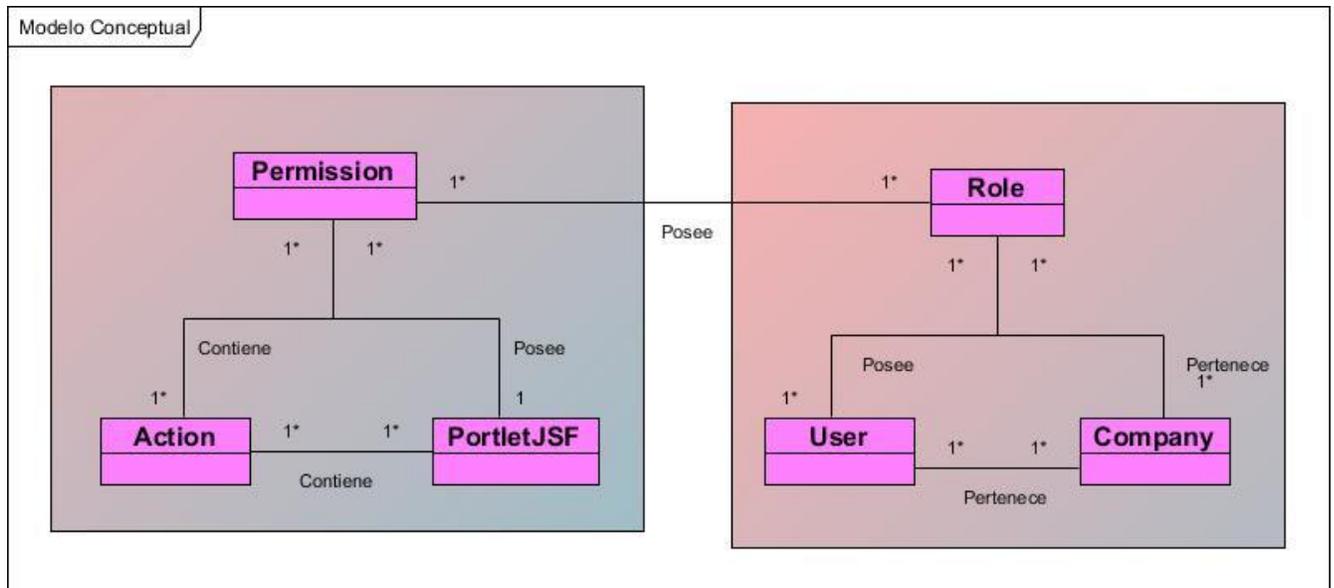


Figura 3 Modelo Conceptual.

2.3.1 Descripción del diagrama de clases del modelo conceptual.

Nombre	Descripción
<b>Permission</b>	Un permiso está definido por las diferentes acciones de grano fino que un rol puede ejecutar sobre un portlet.
<b>Portlet</b>	Es la clase que almacena los portlets existentes en el Liferay Portal.
<b>Action</b>	Es la clase que almacena las acciones de grano fino que se pueden ejecutar en los portlets.
<b>Role</b>	Existen 3 tipos de roles (también llamados ámbitos de los roles): roles de portal, roles de organización y roles de Sitio web. Por defecto se tiene unos roles predefinidos y estos son: <ul style="list-style-type: none"> <li>○ Guest, User, Power User, Administrador.</li> <li>○ Community Administrator, Community member y Community Owner.</li> <li>○ Organization Adminsitrator, Organization Member, Organization Owner.</li> </ul>
<b>User</b>	Son las personas físicas que usan el sistema.

	<ul style="list-style-type: none"> <li>○ Los portales son accedidos por usuarios.</li> <li>○ Los usuarios pueden concentrarse en grupos de usuarios.</li> <li>○ Los usuarios pertenecen a organizaciones.</li> </ul>
<b>Company</b>	Es la organización o entidad a la que pertenecen los usuarios y que poseen un rol determinado.

**Tabla 2 Descripción del diagrama de clases del modelo conceptual**

## 2.4 Modelado de requisitos del sistema

El modelo de requisitos tiene como objetivo delimitar el sistema y capturar la funcionalidad que ofrecerá desde la perspectiva del usuario, garantizando en el futuro la calidad del software.

### 2.4.1 Requisitos funcionales (RF)

No	Nombre	Descripción	Complejidad
[RF1.]	Obtener listado de roles.	El sistema debe permitir obtener los roles existentes en el Liferay Portal.	Baja
[RF2.]	Obtener listado de portlets del Liferay Portal.	El sistema debe permitir obtener los portlets existentes en el Liferay Portal.	Baja
[RF3.]	Gestionar acciones de grano fino.	El sistema debe permitir gestionar las acciones de grano fino que se puedan realizar sobre los portlets, dígase registrar, actualizar o eliminar una acción.	Alta
[RF4.]	Asignar acciones de grano fino a portlet.	El sistema debe permitir asignar acciones de grano fino a los portlets.	Alta
[RF5.]	Registrar portlets.	El sistema debe permitir registrar en la base de datos los portlets con las acciones de grano fino que se puedan realizar sobre éste.	Alta
[RF6.]	Obtener portlet.	El sistema debe permitir obtener los portlets que estén registrados en la base de datos.	Alta

[RF7.]	Listar acciones de grano fino en portlets.	El sistema debe mostrar las acciones pertenecientes al portlet seleccionado.	Alta
[RF8.]	Asignar acciones de grano fino a roles en portlet.	El sistema debe permitir asignar al rol especificado las acciones de grano fino que puede realizar en determinado Portlet.	Alta

**Tabla 3 Requisitos funcionales.**

### 2.4.2 Requisitos no funcionales (RNF)

No	Nombre	Descripción
[RNF1.]	Soporte	El servidor de base de datos será PostgreSQL Server 9.1.x o puede ser MySQL u otro superior a éstos.
[RNF2.]	Soporte	El sistema se ejecutará en los navegadores Mozilla Firefox 20.0, Internet Explorer 7, Chrome 7, Safari u Opera 10 o en otro superior a los mencionados.
[RNF3.]	Soporte	Se usará el servidor web Apache Tomcat 7.0.x.

**Tabla 4 Requisitos no funcionales.**

### 2.5 Historias de Usuario

**Definición:** Una historia de usuario (o user story en inglés) describe una funcionalidad que, por sí misma, aporta valor al usuario. [31]

Se compone de: [31]

- Una descripción escrita de la historia usada como recordatorio y para planificar. (Debe ser breve).
- Conversaciones acerca de la historia que sirven para aclarar los detalles.
- Un criterio de aceptación (idealmente automatizado) que permita determinar cuándo la historia ha sido completada.

#### 2.5.1 Descripción de las Historias de Usuario

**Historia de Usuario**

<b>Número:</b> HU1	<b>Nombre del requisito:</b> Obtener listado de roles.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 1 día
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 8 horas
<b>Descripción:</b> El administrador luego de haber seleccionado el portlet JSF requerido, será capaz de obtener la lista de roles que existe en el Liferay Portal a través del menú selectItem. Una vez seleccionada esta opción se le mostrará la lista de roles existentes en el sistema, pasando a la selección del rol requerido.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

**Tabla 5 HU Obtener listado de roles.**

<b>Historia de Usuario</b>	
<b>Número:</b> HU2	<b>Nombre del requisito:</b> Obtener listado de

	portlets.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 1 día
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 8 horas
<b>Descripción:</b> El administrador será capaz de obtener la lista de portlets que existe en el Liferay Portal a través del menú selectItem. Una vez seleccionada esta opción se le mostrará la lista de portlets existentes en el sistema, pasando a la selección del portlet JSF requerido.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

**Tabla 6 HU Obtener listado de portlets.**

Las demás Historias de usuario se pueden ver en los Anexos.

## 2.6 Estilo arquitectónico

**Sistemas basados en llamado y retorno:** Este estilo arquitectónico permite al arquitecto del sistema construir una estructura de programa relativamente fácil de modificar y ajustar a

escala. Persigue obtener cualidades de escalabilidad y procedimiento remoto, modificabilidad.[32]

### 2.6.1 Arquitectura de la aplicación

Para el desarrollo de la solución se decidió utilizar una arquitectura en 3 Capas, lo que permitirá que las distintas partes de la aplicación se puedan modificar sin que esto afecte las demás capas, es decir, separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio y mecanismos de almacenamiento.

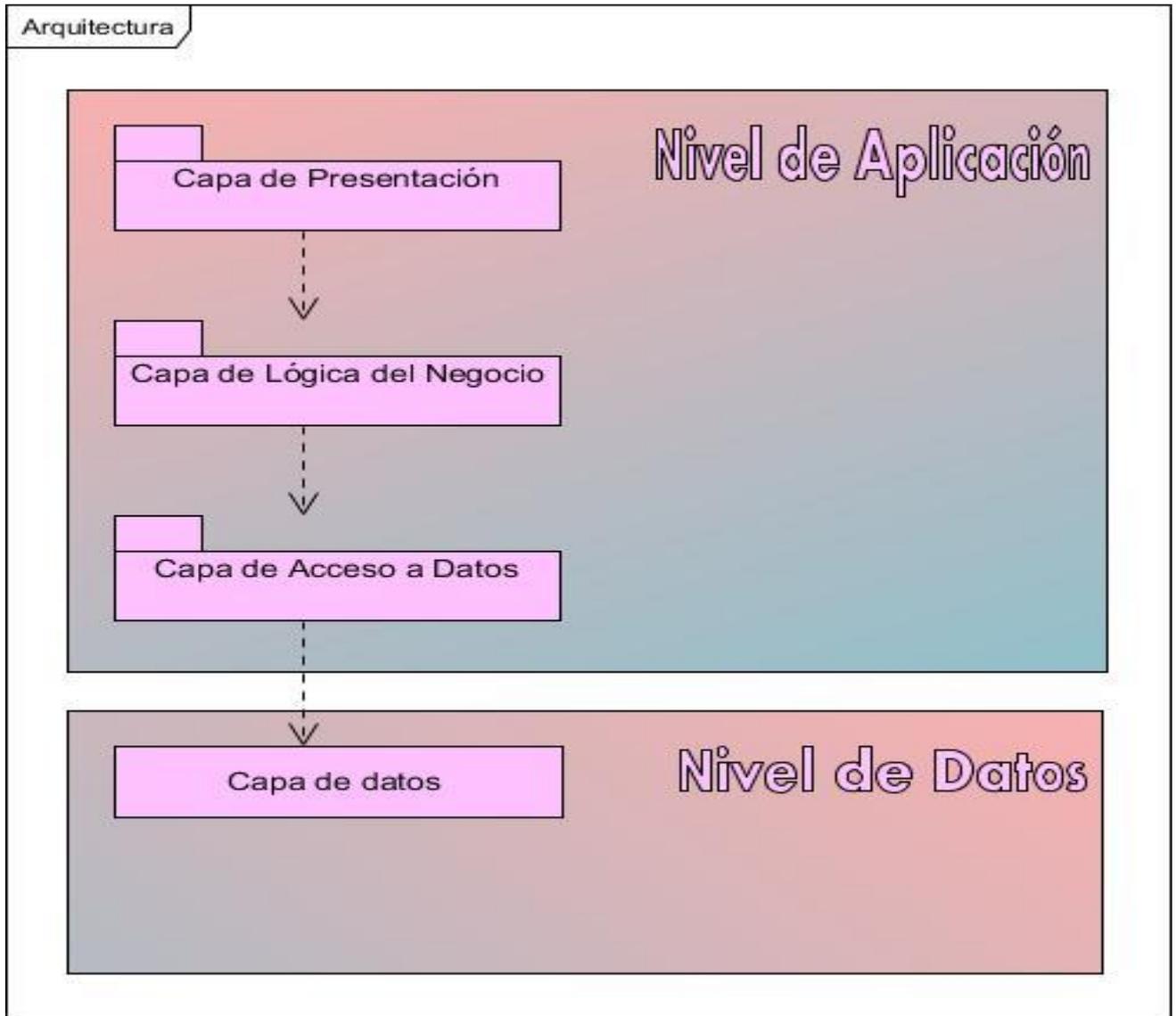


Figura 4 Arquitectura en 3 Capas.

**Capa de presentación:** Se refiere a la presentación del programa frente al usuario, esta presentación debe cumplir su propósito con el usuario final, una presentación fácil de usar y amigable. También las interfaces deben ser consistentes con la información dentro del software (Por ejemplo; en los formularios no debe haber más que lo necesario), tomar en cuenta los requerimientos del usuario, la capa de presentación va de la mano con la capa de la lógica de negocio.[33]

Para el desarrollo de la solución la capa de presentación fue basada en el framework JSF y PrimeFaces, haciendo uso del patrón arquitectónico Modelo Vista Controlador proporcionado por JSF. Esta capa tiene implementados beans y un controlador JSF para facilitar el manejo de las peticiones realizadas por los usuarios. A través del controlador `face_config` se accede a la segunda capa que implementa la lógica de negocio.

**Capa de lógica de negocios:** En esta capa es donde se encuentran los programas que son ejecutados, recibe las peticiones del usuario y posteriormente envía las respuestas tras el proceso. Esta capa es muy importante pues es donde se establecen todas aquellas reglas que se tendrán que cumplir.[33]

La lógica de negocio es la intermediaria entre la capa de presentación y la capa de acceso a datos, haciéndose uso de Spring como mecanismo de comunicación.

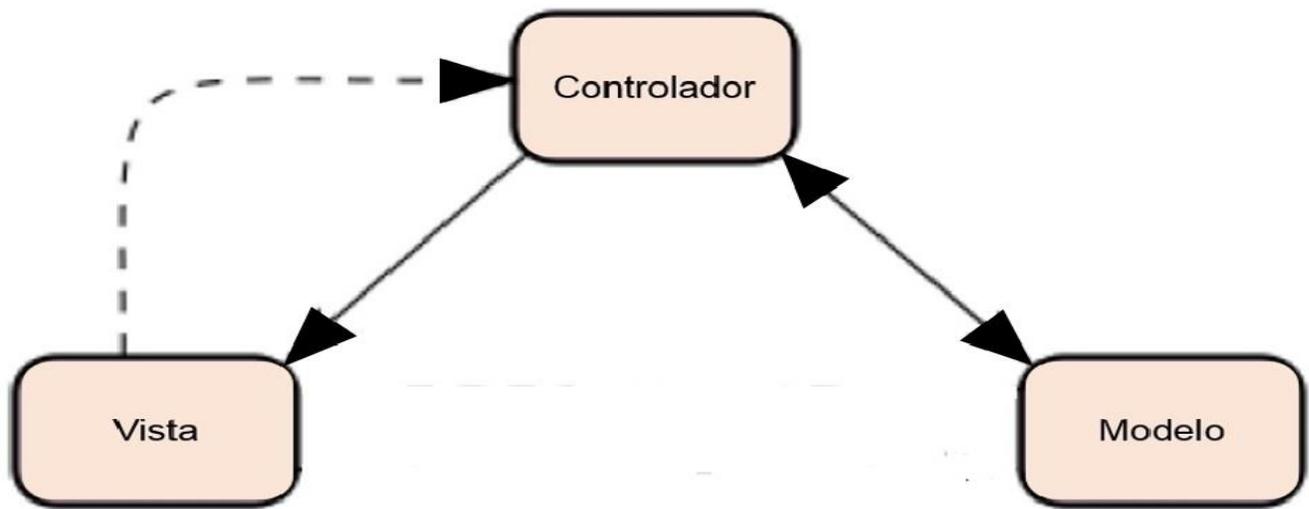
**Capa de datos:** Esta capa es la que se encarga de hacer las transacciones con la base de datos y con otros sistemas para descargar o insertar información al sistema. La consistencia en los datos es sumamente importante, es decir, los datos que se ingresan o insertan deben ser precisos y consistentes.[33] Esta capa mantendrá la comunicación con la de lógica de negocio al enviar la información que será procesada e ingresada en objetos según sea necesario.

La capa de acceso a datos es la encargada de satisfacer las peticiones del negocio a través de los Objetos de Acceso a Datos (DAO), para ello, se utilizan interfaces DAO y sus implementaciones, haciendo uso del framework Hibernate para persistir los datos.

## 2.7 Patrón Modelo Vista Controlador

**Modelo Vista Controlador (MVC):** MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se

potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.[34]



**Figura 5 Patrón Modelo Vista Controlador.**

El **Modelo** representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.[35]

La **Vista** maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo. [35]

El **Controlador** proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.[35]

## 2.8 Patrones de diseño

**Patrón DAO (Data Access Object):** El problema que viene a resolver este patrón es el de contar con diversas fuentes de datos (base de datos, archivos, servicios externos, etc). De tal

forma que se encapsula la forma de acceder a la fuente de datos. Este patrón surge históricamente de la necesidad de gestionar una diversidad de fuentes de información, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma de acceder a éstos.

### 2.8.1 Patrones GRASP

**Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado.[36]

Queda evidenciado el uso del patrón a través del faces-config y los beans manejados, que van a facilitar las peticiones realizadas por los usuarios y que tendrán algún impacto o requieran de una respuesta de la aplicación, además, es a través del faces-config que se accede a la segunda capa que implementa la lógica de negocio.

**Experto en información:** El patrón experto sugiere asignar la responsabilidad al Objeto que posea la información necesaria para desempeñarla.[37]

Queda evidenciado el uso del patrón a través de las clases del modelo: Permission, Portlet, Action, que son las entidades del negocio.

**Alta cohesión:** El patrón Alta Cohesión es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal; una clase con baja cohesión realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar. [38]

Queda evidenciado el uso del patrón a través de los beans: PermissionBean, PortletBean, ActionBean y UtilBean, que serían aquellos que controlan el estado de las páginas web.

**Bajo Acoplamiento:** Es una medida de la fuerza con que una clase se relaciona con otras; es decir, no posee numerosas relaciones.[38]

Queda evidenciado a través de la arquitectura en capas, donde estaría estructurada por paquetes: interfaces, beans, Service, dao, model, evitando el acceso directo a los datos, es decir, que si existe algún cambio en las capas inferiores no afectaría las capas superiores.

### 2.8.2 Patrones GOF

**Facade:** Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar. [39]

Sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas.[39]

**Singleton:** Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. [39]

El patrón de diseño singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. [39]

El uso de los patrones mencionados anteriormente, quedan evidenciados a través de la clase Service, que proporciona un punto de acceso global, manejado por Spring, actuando como mecanismo de comunicación con la capa de acceso a datos la cual es responsable del manejo de los datos.

### 2.9 Modelo de diseño

Diagramas de clases:

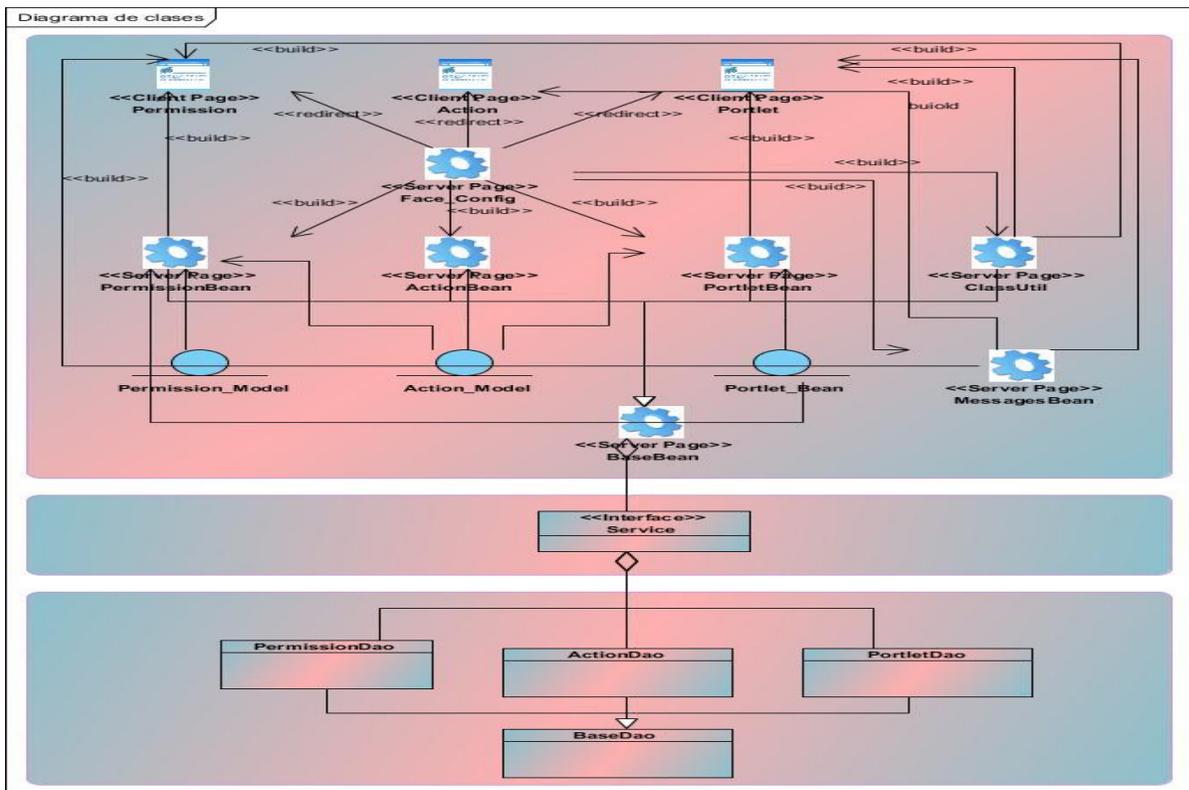


Figura 6 Diagrama de clases del diseño.

### 2.9.1 Descripción de las clases del diseño

Nombre	Descripción	Tipo
<b>Permission</b>	Es la interfaz principal mostrada al usuario al solicitar el portlet que gestiona los permisos asignados a los diferentes roles del sistema en los recursos, permitiéndole navegar por las demás interfaces.	Interfaz
<b>Portlet</b>	Permite registrar los portlets con las acciones de grano fino que se pueden ejecutar en el mismo.	Interfaz
<b>Action</b>	Gestiona las acciones de grano fino que se pueden realizar sobre un portlet.	Interfaz
<b>Service</b>	Esta clase es una interfaz para hacer más fácil el acceso a las funcionalidades implementadas unificando las acciones más comunes, permitiendo comunicar la capa lógica de negocio con la capa de acceso a datos.	Interfaz
<b>faces-config</b>	Es el controlador principal, que facilita el manejo de las peticiones realizadas por los usuarios, además, es el encargado de comunicar la capa de presentación con la capa lógica de negocio.	Controladora
<b>PermissionBean</b>	Esta clase implementa todas las funcionalidades necesarias para construir la vista principal, gestionando de esta forma, los permisos otorgados a los roles sobre los portlets.	Controladora
<b>PortletBean</b>	Esta clase implementa todas las funcionalidades necesarias para construir la vista Portlet, asignándoles las diferentes acciones que se	Controladora

	podrán desarrollar sobre éste y posteriormente almacenarlos.	
<b>ActionBean</b>	Esta clase implementa todas las funcionalidades necesarias para construir la vista Acciones, gestionando las acciones que se pueden realizar sobre un portlet.	Controladora
<b>UtilBean</b>	Esta clase implementa las funcionalidades necesarias para construir las vistas (Principal, Portlet, Acciones), obteniendo datos indispensables del Liferay, que garanticen que el portlet gestione las peticiones requeridas.	Controladora
<b>Permission</b>	Esta clase almacena los permisos otorgados a los roles sobre los portlets.	Modelo
<b>Portlet</b>	Esta clase almacena los portlets y sus respectivas acciones de grano fino.	Modelo
<b>Action</b>	Esta clase almacena todas acciones de grano fino que se puedan ejecutar sobre los portlets.	Modelo

**Tabla 7 Descripción de las clases del diseño**

Las clases con sus respectivos atributos y funcionalidades pueden verse en los anexos.

### 2.11 Conclusiones parciales

- Para un correcto trabajo en el cumplimiento de los requisitos definidos, se estableció la arquitectura que debe tener la aplicación a desarrollar, así como los patrones de diseño que permitirán obtener ventajas significativas en su utilización.
- Se expusieron y explicaron todos los artefactos generados por la metodología AUP verificándose la factibilidad de su uso para dar cumplimiento a las tareas que impuso la solución propuesta.

## Capítulo #3. Implementación y prueba del sistema

### 3.1 Introducción

En el presente capítulo se describe el proceso de implementación de la solución propuesta. Se plasman los casos de pruebas realizados a las historias de usuario correspondientes demostrando de esta forma el cumplimiento de los requerimientos definidos.

### 3.3 Estándar de código

Un Estándar de programación es: Una forma de "normalizar" la programación de forma tal que al trabajar en un proyecto, cualquier persona involucrada en el mismo tenga acceso y comprenda el código.[40]

#### Variables:

Para estos identificadores haremos uso de la variante lowerCamelCase. Empiezan con minúsculas y si estos identificadores están compuestos por varias palabras las siguientes comenzarán con mayúscula.

#### Métodos:

Los nombres de métodos deben iniciar con un verbo.

Ejemplo:

```
public void registerPermission(...) {...}
```

Los métodos para obtener campos privados en las clases tienen el prefijo "get".

Ejemplo:

```
public String getActionId() {...}
```

Los modificadores de campos privados en las clases tienen el prefijo "set".

Ejemplo:

```
public void setActionId(...) {...}
```

Los obtenedores con el resultado de booleano tienen como nombre un verbo.

Ejemplo:

```
public Boolean get(...) {...}
```

#### Constantes:

Las constantes o campos finales son escritos en letras minúsculas.

Ejemplo:

```
portletjsf = new ArrayList<PortletJSF>();
```

### Clases:

Para estos identificadores se hace uso de la variante UpperCamelCase. Todas las palabras que componen a dichos identificadores empezarán con mayúscula.

Ejemplo:

```
public class PortletJSF implements Serializable{...}
```

### Comentarios de las clases:

Al inicio de cada clase se debe describir con un comentario de bloque el propósito de la clase e instrucciones de uso.

Ejemplo:

```
/**
 *
 * El método permite obtener un permiso dado el identificador del rol y el
 * identificador del portlet jsf.
 *
 * @parámetro long idrol
 * el primer valor
 * @parámetro PortletJSF
 * el segundo valor
 * @retorna Permiso
 *
 */
```

### 3.4 Pruebas

Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un sistema. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Las pruebas de software se integran dentro de las diferentes fases del ciclo del software que se ubican dentro de la Ingeniería de Software. Para determinar el nivel de factibilidad y calidad

se le realizan pruebas a la aplicación mediante técnicas experimentales usando pruebas de caja negra.

**Pruebas de caja negra**

- Pruebas que se llevan a cabo sobre la interfaz del software.
- El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se ve el código).

**3.4 Estrategia de pruebas**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Obtener listado de roles.	EC 1.1: Se obtuvo correctamente el listado de roles.	El sistema debe permitir obtener los roles existentes en el Liferay Portal.	1- En el portal Liferay, en el portlet “Roles for Portlets”, accionar el ítem “Seleccionar Role”.  2-Se desplegará la lista de roles existentes en el portal.  3-Se procederá a la selección del rol requerido.
SC 2: Obtener listado de portlets del Liferay Portal.	EC 2.1: Se obtuvo correctamente el listado de portlets.	El sistema debe permitir obtener los portlets existentes en el Liferay Portal.	1- En el portal Liferay, en el portlet “Roles for Portlets”, accionar el ítem “Seleccionar Portlet”.  2-Se procederá a la selección del portlet requerido.

<p>SC 3: Registrar acción de grano fino.</p>	<p>EC 3.1: Acción registrada satisfactoriamente.</p>	<p>El sistema debe permitir registrar en la base de datos las acciones de grano fino que se puedan realizar en los portlets.</p>	<p>1- En el portal Liferay, en el portlet “Roles for Portlets”, dirigirse al menú “Configuración” y seleccionar “Acciones”. Se mostrará un formulario con las acciones existentes.</p> <p>2-Aparecerá un formulario para llenar con los datos de la acción que se quiera ingresar.</p> <p>3-Luego de introducir los datos requeridos seleccionará el botón “Guardar”, en caso de que exista un campo vacío el sistema notificará el campo en cuestión y mostrará el mensaje “El campo es requerido” o se introducen caracteres inválidos el sistema mostrará el mensaje “Solo introducir letras”.</p> <p>4-Al concluir la operación el sistema mostrará el mensaje “Acción guardada exitosamente”.</p>
<p>SC 4: Actualizar acción de grano fino.</p>	<p>EC 4.1: Acción actualizada satisfactoriamente.</p>	<p>El sistema debe permitir actualizar en la base de datos las acciones que</p>	<p>1- En el portal Liferay, en el portlet “Roles for Portlets”, dirigirse al menú “Configuración” y seleccionar “Acciones”. Se mostrará un</p>

		<p>se pueden realizar en los portlets.</p>	<p>formulario con las acciones existentes.</p> <p>2-Seleccionar la acción que se desea modificar.</p> <p>3-Aparecerá un formulario con los datos de la acción.</p> <p>4-Luego de introducir los datos requeridos seleccionará el botón “Actualizar”, en caso de que exista un campo vacío el sistema notificará el campo en cuestión y mostrará el mensaje “El campo es requerido” o se introducen caracteres inválidos el sistema mostrará el mensaje “Solo introducir letras”.</p> <p>5-Al concluir la operación el sistema mostrará el mensaje “Acción actualizada exitosamente”.</p>
<p>SC 5: Eliminar de grano fino acción.</p>	<p>EC 5.1: Acción eliminada satisfactoriamente.</p>	<p>El sistema debe permitir eliminar en la base de datos las acciones que se pueden realizar en los Portlets JSF.</p>	<p>1- En el portal Liferay, en el portlet “Roles for Portlets”, dirigirse al menú “Configuración” y seleccionar “Acciones”. Se mostrará un formulario con las acciones existentes.</p>

			<p>2-Seleccionar el nombre de la acción que se desea eliminar, y accionar el botón “Eliminar”, el sistema mostrará el mensaje “Esta segura que desea eliminar permanentemente el elemento seleccionado”, en caso contrario puede optar por el botón “Cancelar”.</p>
<p>SC 6: Asignar acciones de grano fino a portlet.</p>	<p>EC 6.1: Asignación satisfactoria de acciones a portlet.</p>	<p>El sistema debe permitir asignar acciones de grano fino a los portlets.</p>	<p>1- En el portal Liferay, en el portlet “Roles for Portlets”, dirigirse al menú “Configuración” y seleccionar “Portlets”.</p> <p>2-Seleccionar el portlet al que se quiere agregar las acciones.</p> <p>3-Accionar el ítem “Seleccionar las acciones”, al seleccionarse las acciones requeridas, el sistema mostrará la lista de acciones seleccionadas.</p>
<p>SC 7: Registrar portlets.</p>	<p>EC 7.1: Portlet registrado satisfactoriamente.</p>	<p>El sistema debe permitir registrar en la base de datos el</p>	<p>1-En el portal Liferay, en el portlet “Roles for Portlets”, dirigirse al menú</p>

		portlet seleccionado.	<p>“Configuración” y seleccionar “Portlets”.</p> <p>2-Seleccionar el portlet al que se quiere agregar las acciones.</p> <p>3-Accionar el ítem “Seleccionar las acciones”, al seleccionarse las acciones requeridas, el sistema mostrará la lista de acciones seleccionadas.</p> <p>4-Luego de seleccionado los datos requeridos, accionar el botón “Guardar”,</p> <p>5-Al concluir la operación el sistema mostrará el mensaje “Portlet guardado exitosamente”.</p>
SC 8: Obtener listado de portlets.	EC 8.1: Se obtuvo correctamente el listado de portlets.	El sistema debe permitir obtener los portlets desarrollados con que estén registrados en la base de datos.	<p>1-En el portal Liferay, en el portlet “Roles for Portlets”, accionar el ítem “Seleccione el portlet”.</p> <p>2-Se desplegará la lista de portlets existentes en la base de datos.</p> <p>3-Se procederá a la selección del portlet requerido.</p>

<p>SC 9: Listar acciones de grano fino en portlets.</p>	<p>EC 9.1: Se listaron las acciones del portlet correctamente.</p>	<p>El sistema debe mostrar las acciones de grano fino pertenecientes al portlet seleccionado.</p>	<p>1-En el portal Liferay, en el portlet “Roles for Portlets”, accionar el ítem “Seleccionar Portlet”.</p> <p>2-Luego de seleccionado el portlet requerido, el sistema mostrará las acciones del mismo en el campo “Permiso”.</p>
<p>SC 10: Asignar acciones de grano fino a roles en portlet.</p>	<p>EC 10.1: Asignación satisfactoria de acciones de grano fino a roles en portlet.</p>	<p>El sistema debe permitir asignar al rol especificado las acciones de grano fino que puede realizar en determinado Portlet.</p>	<p>1-En el portal Liferay, en el módulo “Roles for Portlets”, accionar el ítem “Seleccionar Portlet”.</p> <p>2-Luego de seleccionado el portlet requerido, el sistema mostrará las acciones del mismo en el campo “Permiso”.</p> <p>3-Luego de la operación, el sistema permitirá accionar en el ítem “Seleccionar el rol” y optar por el rol requerido.</p> <p>4-Luego de seleccionado el rol requerido, el sistema mostrará las acciones que éste puede realizar sobre el portlet en cuestión en el campo “Autorizado”.</p>

			<p>5-Luego se podrá optar por otorgarle o quitarle permisos sobre este recurso.</p> <p>6-Precionar el botón “Guardar”, mostrándose el mensaje “Permiso guardado exitosamente”.</p>
--	--	--	--

**Tabla 8 Estrategia de pruebas**

**Técnica de la Partición de Equivalencia**

La partición de equivalencia es un método de prueba de Caja Negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.[41]

En otras palabras, este método intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia.[41]

De tal modo que se pueda asumir razonablemente que una prueba realizada con un valor representativo de cada clase es equivalente a una prueba realizada con cualquier otro valor de dicha clase. Esto quiere decir que si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error. Y viceversa, si un caso de prueba no ha detectado ningún error, es de esperar que ninguno de los casos de prueba correspondientes a la misma clase de equivalencia encuentre ningún error.[41]

**3.5 Casos de pruebas**

Las celdas de la tabla contienen **V**, **I**, o **N/A**. **V** indica válido, **I** indica inválido, y **N/A** que no es necesario proporcionar un valor del dato en este caso, dado que es irrelevante.

Id del escenario	Escenario	Variable 1 Nombre	Variable2 Descripción	Respuesta del Sistema	Resultado de la Prueba
------------------	-----------	----------------------	--------------------------	-----------------------	------------------------

EC 1	Registrar acción.	V Adicionar__Portlet	V Permite adicionar los portlets.	Mensaje de confirmación	Se agrega una acción al sistema.
		I Nulo o números	V Permite adicionar los portlets	Notificación de error	
		V Adicionar__Portlet	I Nulo o números	Notificación de error	
		I Nulo o números	I Nulo o números	Notificación de error	
EC 2	Modificar acción.	V Adicionar_Acción	V Permite adicionar acciones al sistema.	Mensaje de confirmación	Se actualizó satisfactoriamente la acción.
		I Nulo o números	V Permite adicionar acciones al sistema.	Notificación de error	

		V Adicionar_Acción	I Nulo o números	Notificación de error	
		I Nulo o números	I Nulo o números	Notificación de error	
EC 3	Eliminar acción.	N/A	N/A	Se elimina la consulta correctamente.	Se elimina la consulta seleccionada y previamente registrada en el sistema.

**Tabla 9 Caso de prueba EC 1: Registrar acción**

Los demás Casos de pruebas se pueden ver en los Anexos.

### 3.6 Lista de chequeo [42]

Característica de soporte de comprobar.	Ponderación de importancia	Nivel de cumplimiento	Justificación
1-¿Se utilizó como servidor de base de datos PostgreSQL 9.1 para el desarrollo del sistema?	4 1(poco), (fundamental)	5 (0-100)	Se utilizó el gestor de base de datos PostgreSQL 9.1, por ser un requisito establecido por el cliente, además, es uno de los más usados en la Universidad de Ciencias Informáticas.

			Debido a la liberación de la licencia. El código fuente se encuentra disponible para todos sin costo alguno.
¿El Portlet Administrativo es adaptable a los navegadores Moxilla Firefox 20.0, Internet Explorer 7, Chrome 7, Safari u Opera?	5	100	El sistema se ejecutó en diferentes entornos de navegación: Moxilla Firefox, Internet Explorer, Chrome, Safari y Opera.
¿Para desarrollar la aplicación se utilizó como servidor web el Apache Tomcat 7.0?	5	100	Se utilizó el servidor web el Apache Tomcat por ser requerido por el cliente.  Posee ayuda y abundante documentación.  Cuenta con una comunidad de usuarios, es libre y uno de los más usados en el CDAE.  Para trabajar con esta herramienta se instaló la Máquina Virtual de Java.

**Tabla 9 Lista de chequeo**

### 3.5.1 Resultados

Se realizaron dos iteraciones de las pruebas de caja negra específicamente con la utilización de casos de pruebas y técnica de partición de equivalencia, para comprobar que la solución funcionara correctamente y si satisfacía los requerimientos del cliente. Durante la primera

iteración se encontraron siete no conformidades, una de prioridad Alta y seis de prioridad Baja. La no conformidad de prioridad Alta fue con respecto a mal funcionamiento de validación de la aplicación y las Bajas estuvieron relacionadas con mal funcionamiento de mensajes de confirmación, todas fueron resueltas por la desarrolladora de la aplicación. Se realizó una segunda iteración, en la cual no se encontraron no conformidades, evaluándose de esta manera todos los casos de prueba como satisfactorios.

### **3.6 Conclusiones parciales**

- En el capítulo para concretar la implementación y prueba del sistema, se presentaron los casos de pruebas.
- Las pruebas que se le realizaron al sistema trajeron consigo mejoras en la implementación de las capas de la lógica del negocio y presentación, logrando una apariencia más amigable con el cliente.

## **Conclusiones generales**

- Las herramientas y tecnologías puestas en práctica permitieron el desarrollo claro y fluido de un sistema construido sobre bases sólidas y un entorno bien definido.
- Se validó la solución mediante el uso de pruebas de aceptación, a través de prueba de caja negra arrojando como resultado el buen funcionamiento del portlet desarrollado, quedando de esta forma resuelto el problema investigativo.

## **Recomendaciones**

- Continuar con el proceso de desarrollo del portlet con el objetivo de implementar nuevas funcionalidades.

## Bibliografía Referenciada

1. Rosas, J.E.S., El Mundo es Open Source Blog sobre productos y soluciones de software libre y código abierto de nivel empresarial, in Liferay: Portales web para el trabajo colaborativo de empresas. 2010 [cited 2015], antartec.com.
2. Liferay Portal 6.2 mejora la experiencia web y dota de más protagonismo a la movilidad. [Página Web] 2013 [cited 2015]; Available from: <http://www.comunicacioneshoy.es/es/liferay-portal-62-mejora-la-experiencia-web-y-dota-de-mas-protagonismo-la-movilidad/23/10/2013>.
3. Reservados, L.I.T.I.D. Use Liferay para construir modernos sitios web y portales para fidelizar a sus clientes, partners y empleados. 2015 [cited 2015]; Available from: <http://www.liferay.com/es/>.
4. Alegsa, L. DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA . Definición de Seguridad informática [Página Web] 2015 [cited 2015]; Available from: <http://www.alegsa.com.ar/Dic/seguridad%20informatica.php>.
5. Mifsud, E. Introducción a la seguridad informática - Mecanismos básicos de seguridad. [Página Web] 2012 [cited 2015]; Available from: <http://recursostic.educacion.es/observatorio/web/es/component/content/article/1040-introduccion-a-la-seguridad-informatica?start=2>.
6. Política de seguridad. [Página Web] 2015 [cited 2015]; Available from: <http://www.guiaspracticas.com/controles-de-acceso/politica-de-seguridad>.
7. Infante, A.C. and A.H. González Control de Acceso. 2014 [cited 2015].
8. documentation, L.w.p.v., Control de Acceso. latinix 2015 [cited 2015].
9. Solaris, O. Guía de administración del sistema: servicios de seguridad 2011 [cited 2015]; Available from: [http://docs.oracle.com/cd/E24842\\_01/html/E23286/rbac-1.html](http://docs.oracle.com/cd/E24842_01/html/E23286/rbac-1.html).
10. Hernández, I.C., Plataforma de Aprendizaje a Distancia Basada en tecnología de Portlets y Web 2.0 para una Enseñanza Participativa. Sistemas Informáticos,2008 [cited 2015]: p. 1-97.
11. liferay.com. Liferay Portal [Página Web] 2015 [cited 2015]; Available from: <http://www.liferay.com/es/products/liferay-portal/overview>.
12. Scamarcio, F. Introducción a Liferay Portal. [Página Web.] 2010 [cited 2015]; Available from: <http://francescoscamarcio.com/2010/12/10/introduccion-a-liferay-portal/>.
13. Corporation, C.I. Conceptos de portlets. [Página Web] 2010 [cited 2015]; Available from: [http://www-01.ibm.com/support/knowledgecenter/SSYJ99\\_6.1.0/com.ibm.wp.exp.doc\\_v6101/dev/wpsbp.c.html?lang=es](http://www-01.ibm.com/support/knowledgecenter/SSYJ99_6.1.0/com.ibm.wp.exp.doc_v6101/dev/wpsbp.c.html?lang=es).
14. Casas, A.A.C., Desarrollo de un Software que implemente un modelo de madurez para SOA., in Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE). Universidad de las Ciencias Informáticas. p. 78.

15. Fernández, O.B., Introducción al lenguaje de programación Java, in Una guía básica. 2005 [cited 2015], uji.es.
16. Arquitectura, análisis y desarrollo de software empresarial: Dynamics NAV, QlikView y JavaEE. Vaadin – Aplicaciones web Java como si fuesen de escritorio. [Página Web] 2012 [cited 2015]; Available from: <http://carlospesquera.com/vaadin-aplicaciones-web-java-como-si-fuesen-de-escritorio/>.
17. García, A.P. Introducción a JSF (Java Server Faces). Primer artículo de un pequeño manual sobre esta tecnología. [Página Web] 2006 [cited 2015]; Available from: <http://www.desarrolloweb.com/articulos/2380.php>.
18. Almirón, C.G. Introducción a JSF Java Server Faces. [Página Web] 2009 [cited 2015]; Available from: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>.
19. Cruz Quispe Víctor Fabio, G.M.E.D., Mendivil Torrico Luís Briam, Diagrama de componentes. 2011 [cited 2015], LinkedIn Corporation.
20. Lerma, E.V. Introducción a Primefaces. [Página Web] 2010 [cited 2015]; Available from: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=introduccionPrimefaces>.
21. Briano, F. Introducción a Spring Framework Java [Página Web] 2010 [cited 2015]; Available from: <http://picandocodigo.net/2010/introduccion-a-spring-framework-java/>.
22. Fulguera, J.O., Spring Framework. 2014 [cited 2015].
23. EducacionIT. ¿Qué es Java Hibernate? [Página Web] 2013 [cited 2015]; Available from: <http://blog.educacionit.com/2013/02/07/que-es-java-hibernate/>.
24. Dev, G. Eclipse IDE. [Académico] 2014 [cited 2015]; Available from: <http://www.genbetadev.com/herramientas/eclipse-ide>.
25. Romero, H., Metodologías de desarrollo. 2012 [cited 2015], es.slideshare.net.
26. Nieves, I.I.S., METODOLOGÍAS ÁGILES. Proceso Unificado Ágil (AUP). Ingeniería del Software II – Análisis de Sistemas, 2014 [cited 2015]: p. 9.
27. Sánchez, T.R. PROGRAMA DE MEJORA Metodología de desarrollo para la Actividad productiva de la UCI. 2014 [cited 2015].
28. SADRADÍN, L.C., SISTEMA AUTOMATIZADO PARA EL DISEÑO DE PROTOTIPOS DE INTERFACES DE USUARIO. 2009 [cited 2015], Universidad de las Ciencias Informáticas. p. 102.
29. González, L.C. and E.R.P. Torres, Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. Serie Científica de la Universidad de las Ciencias Informáticas, 2012 [cited 2015]. 5: p. 11.
30. Modelos conceptuales. [Página web] 2014 [cited 2015]; Available from: <http://www.buenastareas.com/ensayos/Modelos-Conceptuales/23455.html>.

31. Beas, J.M. Historias de Usuario. [Página web] 2011 [cited 2015]; Available from: <http://jmbeas.es/guias/historias-de-usuario/>.
32. Cham, F., Introducción al diseño arquitectónico. 2013 [cited 2015], Tecnología.
33. Zamora, C.E.R., La programación por capas MilkZoft, Editor. 2014 [cited 2015], CODEJOBS Aprende a programar.
34. Álvarez, M.A. Te explicamos de manera general MVC, Model - View - Controller o Modelo - Vista - Controlador un patrón de diseño de software para programación que propone separar el código de los programas por sus diferentes responsabilidades. [Página Web] 2014 [cited 2015]; Available from: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>.
35. Romero, Y.F. and Y.D. González, Patrón Modelo-Vista-Controlador. 2012 [cited 2015]. **11**: p. 11.
36. Carmona, J.G., in GRASP: Controlador. 2012 [cited 2015], Juan García Carmona.
37. Rey, E.M., Diseño Dirigido por Responsabilidades con los patrones GRASP. 2015 [cited 2015], Pearson Educación, S.A. Todos los derechos reservados.
38. Tabares, R.B. Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos desde Lógica de Programación. [Libro electrónico] 2011 [cited 2015]; Available from: <http://biblioteca.ucp.edu.co/ojs/index.php/entrecei/article/view/760/721>.
39. Mühlrad, D., Patrones de diseño. 2008 [cited 2015].
40. Emanuel, Estándares y Buenas Practicas. 2013 [cited 2015], es.slideshare.net.
41. Natalia Juristo, A.M.M., Sira Vegas TÉCNICAS DE EVALUACIÓN DE SOFTWARE. 2006 [cited 2015], grise.upm.es.
42. ACOSTA, A.O. and L.M.C. TAPASCO, IMPACTO DE LAS PRUEBAS NO FUNCIONALES EN LA MEDICIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE DESARROLLADO, in PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN PEREIRA. 2011 [cited 2015], UNIVERSIDAD TECNOLÓGICA DE PEREIRA FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y CIENCIAS DE LA COMPUTACION p. 74.

## Anexos

Historia de Usuario													
<b>Número:</b> HU3	<b>Nombre del requisito:</b> Gestionar acciones de grano fino.												
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1												
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 10 días												
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 40 horas												
<p><b>Descripción:</b> El administrador al elegir la opción “Acciones” al haberla seleccionado en el menú “Configuración” , el sistema mostrará una página con las acciones existentes, permitiendo optar por registrar una nueva acción a través del botón “Acciones”, el sistema mostrará un formulario para introducir los datos requeridos, en este caso: el nombre y la descripción de la misma y para finalizar la operación seleccionará el botón “Guardar”, que permitirá guardar la nueva acción en la base de datos y enviando un mensaje de confirmación “Se ha registrado la acción satisfactoriamente”, almacenándose la siguiente información:</p> <table border="1"> <thead> <tr> <th>Campo</th> <th>Tipo de dato</th> <th>Reglas o restricciones</th> </tr> </thead> <tbody> <tr> <td>accion_id</td> <td>número entero</td> <td>No nulo</td> </tr> <tr> <td>accion_nombre</td> <td>Cadena de caracteres</td> <td>No nulo</td> </tr> <tr> <td>accion_descripcion</td> <td>Cadena de caracteres</td> <td>No nulo</td> </tr> </tbody> </table>		Campo	Tipo de dato	Reglas o restricciones	accion_id	número entero	No nulo	accion_nombre	Cadena de caracteres	No nulo	accion_descripcion	Cadena de caracteres	No nulo
Campo	Tipo de dato	Reglas o restricciones											
accion_id	número entero	No nulo											
accion_nombre	Cadena de caracteres	No nulo											
accion_descripcion	Cadena de caracteres	No nulo											
<p>Si exista la acción y se desea modificar, tendrá la opción de optar por el botón “Actualizar”, que permitirá guardar y validar los cambios realizados y notificando el nuevo cambio, en caso de que se quiera eliminar una acción existente en el sistema, podrá seleccionar el nombre de la acción deseada y se optará por el botón “Eliminar”, el sistema mostrará el mensaje “Esta segura que desea eliminar permanentemente el elemento seleccionado”, en caso contrario puede optar por “Cancelar”.</p>													

**Observaciones:**

**Prototipo de interfaz:**

El prototipo muestra un menú 'Gestionar Acciones' que abre una ventana con el mismo título. La ventana contiene una tabla con las siguientes acciones:

Nombre	Descripción
Registrar	Registra la información.
Actualizar	Actualiza la información.
Eliminar	Elimina la información.

A la derecha de la tabla hay un botón 'Adicionar' y tres pares de botones 'Actualizar' y 'Eliminar' correspondientes a cada fila de la tabla.

**Anexo 1 HU Gestionar acciones de grano fino.**

Historia de Usuario	
<b>Número:</b> HU4	<b>Nombre del requisito:</b> Registrar portlets.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 5 días
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 30 horas
<p><b>Descripción:</b> El administrador será capaz de registrar los portlets, al optar por "Portlets" en el menú "Configuración", mostrándosele una nueva interfaz, donde podrá seleccionarlo en la lista de portlets existentes en el Liferay, luego le asignará las acciones que se podrán desarrollar sobre éste, al finalizar la operación accionará el botón "Registrar". El sistema le enviará un mensaje de confirmación "El portlet se ha registrado satisfactoriamente", en caso de no seleccionar al menos una acción, se notificará con el mensaje "Debe seleccionar una acción", en caso de no haberse seleccionado el portlet, se notificará con el mensaje "Debe seleccionar</p>	

un portlet”.

**Observaciones:**

**Prototipo de interfaz:**



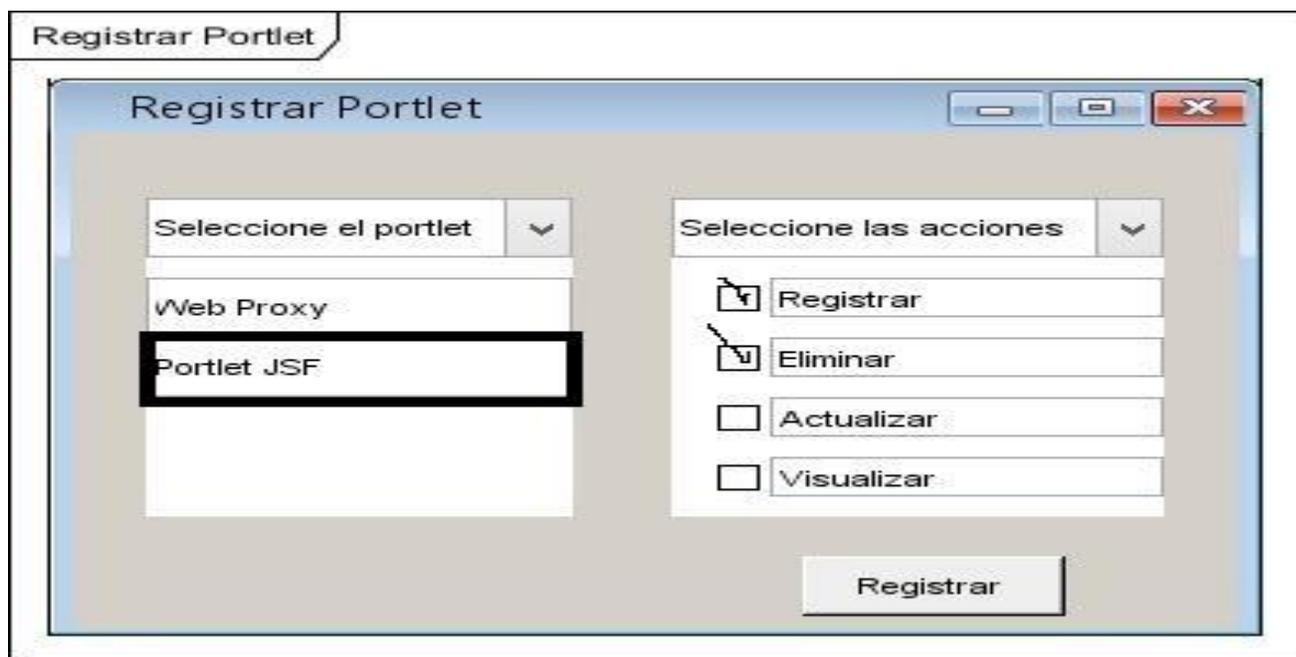
**Anexo 2 HU Registrar portlets.**

Historia de Usuario	
<b>Número:</b> HU5	<b>Nombre del requisito:</b> Asignar acciones de grano fino a portlet.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 5 días
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 20 horas
<b>Descripción:</b> El administrador podrá asignar diferentes acciones de grano fino a los portlets, seleccionándolas de la lista de acciones que se muestra, luego de haber optado por el portlet	

en cuestión, para terminar la operación accionará el botón “Registrar”, que permitirá almacenar la lista de acciones que se podrán realizar sobre el portlet especificado.

**Observaciones:**

**Prototipo de interfaz:**



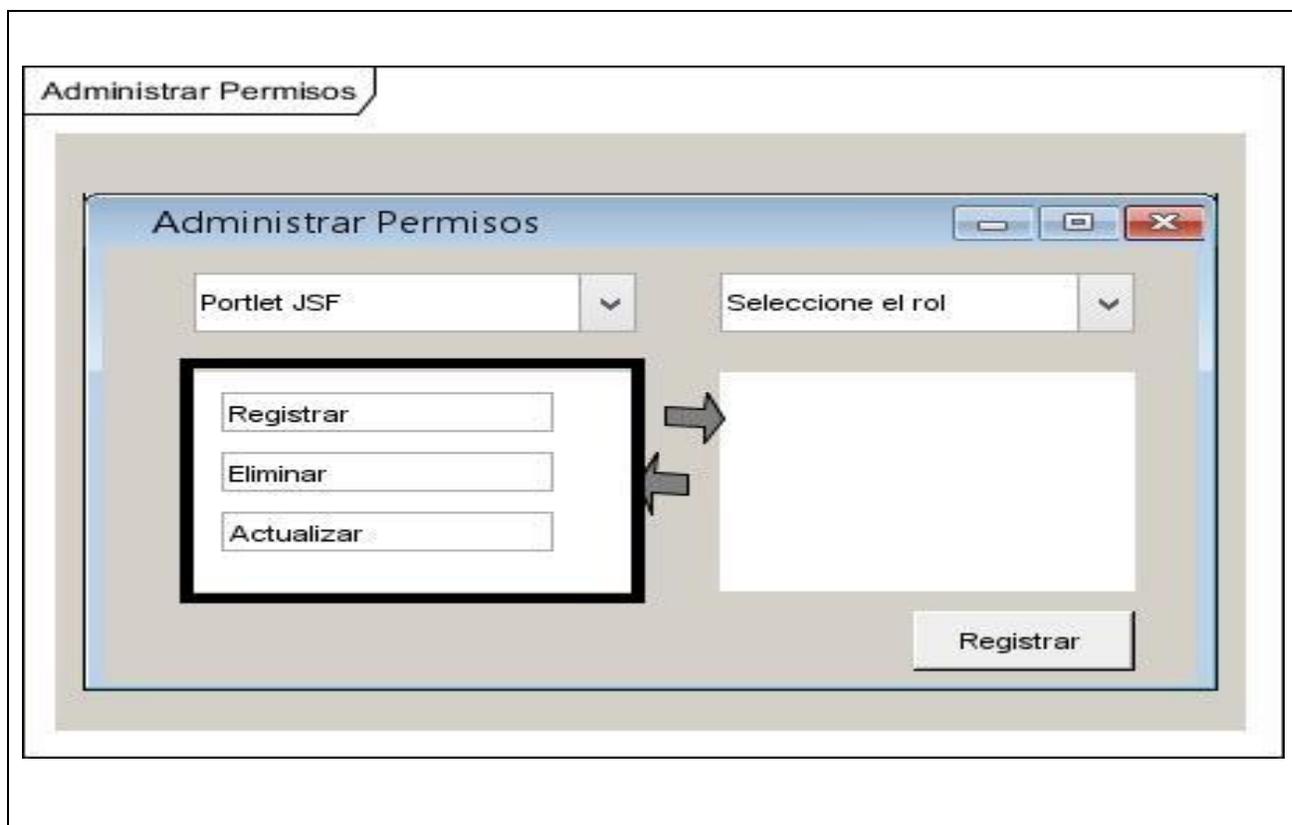
**Anexo 3 HU Asignar acciones de grano fino a portlet.**

Historia de Usuario	
<b>Número:</b> HU6	<b>Nombre del requisito:</b> Obtener Portlet.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 8 horas
<b>Descripción:</b> El usuario obtendrá la lista de portlets al optar por la opción del menú selectItem que desplegará la lista de portlets que se registraron en la base de datos.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	



**Anexo 4 HU Obtener Portlet.**

Historia de Usuario	
<b>Número:</b> HU7	<b>Nombre del requisito:</b> Listar acciones de grano fino en los portlets.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 5 días
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 30 horas
<b>Descripción:</b> El usuario al seleccionar el portlet de la lista de portlets almacenados en la base de datos se le mostrará en el campo "Permisos" las acciones de grano fino que se podrán realizar sobre éste recurso.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	



**Anexo 5 HU Listar acciones de grano fino en los Portlets.**

Historia de Usuario	
<b>Número:</b> HU8	<b>Nombre del requisito:</b> Asignar acciones de grano fino a roles en portlet.
<b>Programador:</b> Susana Alba Silot	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 5 días
<b>Riesgo en Desarrollo:</b> <ul style="list-style-type: none"> <li>○ Problemas eléctricos.</li> <li>○ Problemas técnicos de los servicios de redes.</li> </ul>	<b>Tiempo Real:</b> 40 horas
<b>Descripción:</b> El administrador podrá asignarle o quitarle diferentes acciones de grano fino al rol seleccionado en un portlet determinado. Al seleccionar el componente modular y el rol se mostrarán las acciones a las que tiene permiso en el campo que tiene por nombre "Autorizados" y en "Permisos" quedarán las demás acciones que se pueden realizar sobre el recurso, de esta manera podrá mantener los mismos permisos, adicionarle o quitarle,	

para terminar la operación podrá optar por el botón “Guardar”, que permitirá almacenar la lista de acciones que este rol podrá realizar sobre el portlet en cuestión.

**Observaciones:**

**Prototipo de interfaz:**



**Anexo 6 HU Asignar acciones de grano fino a roles en portlet.**

Id del escenario	Escenario	Variable 1 Lista de roles.	Respuesta del Sistema	Resultado de la Prueba
EC 1	Obtener listado de roles del Liferay Portal.	N/A		El sistema obtiene la lista de roles existentes en el Liferay permitiendo seleccionar el nombre del rol en cuestión.

**Anexo 7 Caso de prueba: EC 1 Obtener listado de roles del Liferay Portal.**

Id del escenario	Escenario	Variable 1 Lista de portlet.	Respuesta del Sistema	Resultado de la Prueba
------------------	-----------	---------------------------------	-----------------------	------------------------

EC 2	Obtener listado de portlets del Liferay Portal.	N/A		El sistema obtiene la lista de portlets existentes en el Liferay permitiendo seleccionar el nombre del portlet en cuestión.
------	---	-----	--	---

**Anexo 8 Caso de prueba: EC 2 Obtener listado de portlets del Liferay Portal.**

Id del escenario	Escenario	Variable 1 Nombre del portlet.	Variable2 Lista de acciones.	Respuesta del Sistema	Resultado de la Prueba
EC 4	Asignar acciones de grano fino a portlet.	V Portlet	V Adicionar_E estudiante Eliminar		El sistema permite asignar acciones de grano fino a los portlets.
		I Nulo	V		
		V	I Nulo		
		I Nulo	I Nulo		

**Anexo 9 Caso de prueba: EC 4 Asignar acciones de grano fino a portlet.**

Id del escenario	Escenario	Variable 1 Nombre del portlet.	Variable2 Lista de acciones.	Respuesta del Sistema	Resultado de la Prueba
------------------	-----------	-----------------------------------	---------------------------------	-----------------------	------------------------

EC 5	Registrar portlet.	V Portlet	V Adicionar_E estudiante Eliminar	Mensaje de confirmación	El sistema registra un portlet con sus respectivas acciones de grano fino.
		I Nulo	V Adicionar_E estudiante Eliminar	Notificación de error	
		V Portlet JSF	I Nulo	Notificación de error	
		I Nulo	I Nulo	Notificación de error	

**Anexo 10 Caso de prueba: EC 5 Registrar portlet.**

<b>Id del escenario</b>	<b>Escenario</b>	<b>Variable 1</b> Lista de portlets.	<b>Variable 1</b> Lista de acciones.	<b>Respuesta del Sistema</b>	<b>Resultado de la Prueba</b>
EC 6	Obtener listado de portlets.	N/A	N/A	Mostrar la lista de portlets.	El sistema muestra el listado de portlets, con sus respectivas acciones de grano fino.

**Anexo 11 Caso de prueba: EC 6 Obtener listado de portlets.**

<b>Id del escenario</b>	<b>Escenario</b>	<b>Variable 1</b> Nombre del portlet.	<b>Variable2</b> Lista de acciones.	<b>Respuesta del Sistema</b>	<b>Resultado de la Prueba</b>
EC 7	Listar acciones en portlets.	V Portlet	V Adicionar_E estudiante Eliminar	Mensaje de confirmación	El sistema luego de haberse seleccionado el portlet, muestra las acciones que se pueden realizar sobre éste.
		I Nulo	V Adicionar_E estudiante Eliminar	Notificación de error	
		V Portlet	I Nulo	Notificación de error	
		I Nulo	I Nulo	Notificación de error	

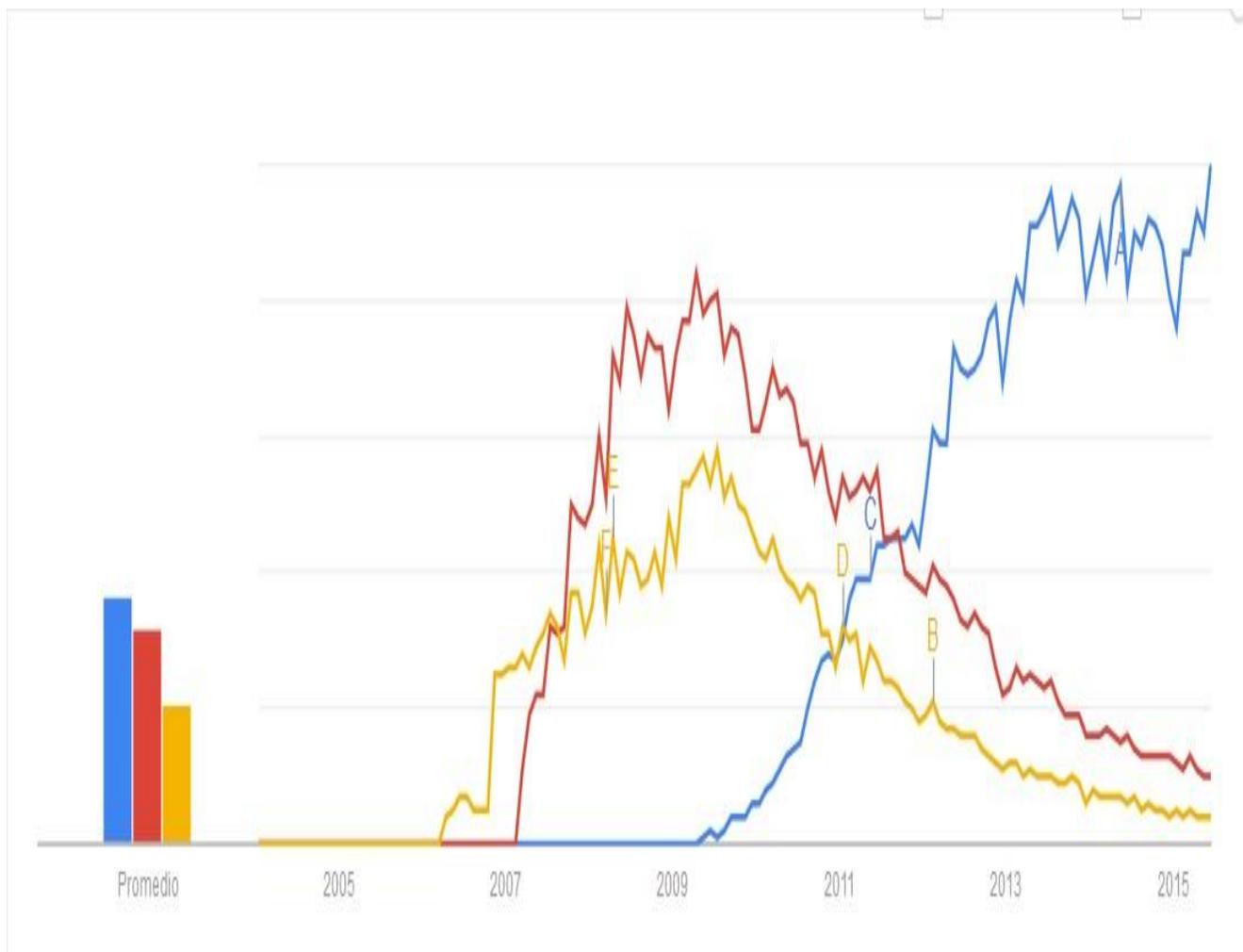
**Anexo 12 Caso de prueba: EC 7 Listar acciones en portlets.**

<b>Id del escenario</b>	<b>Escenario</b>	<b>Variable 1</b> Nombre del portlet.	<b>Variable 2</b> Lista de acciones.	<b>Variable 3</b> Nombre del rol.	<b>Variable 4</b> Lista de acciones	<b>Respuesta del Sistema</b>	<b>Resultado de la Prueba</b>
-------------------------	------------------	--	---	--------------------------------------	--	------------------------------	-------------------------------

					permi tidas.		
EC 8	Asignar acciones a roles en portlet.	V Portlet	N/A	V Administrador_Acción	V Registrar_Acción	Mensaje de confirmación	El sistema guarda satisfactoriamente el permiso asignado al rol especificado sobre el portlet en cuestión.
		I Nulo	V Adicionar_ Estudiante Eliminar	N/A	N/A	Notificación de error.	
		V Portlet	I Nulo	V Administrador_Acción	V Registrar_Acción	Notificación de error.	
		V Portlet	V Adicionar_	I Nulo	V Registrar_	Notificación de error.	

			Estud iante		Acció n	
			Elimi nar			

**Anexo 13 Caso de prueba: EC 8 Asignar acciones de grano fino a roles en portlet.**

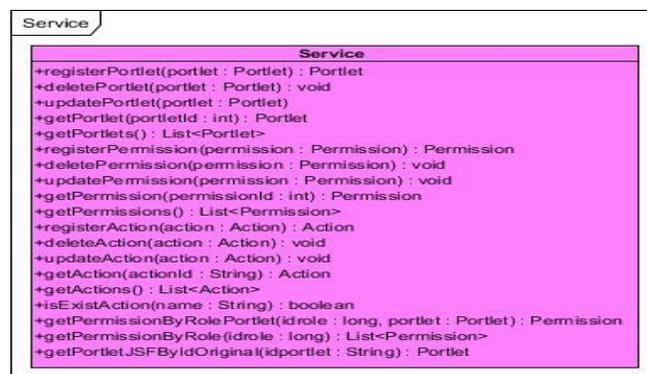
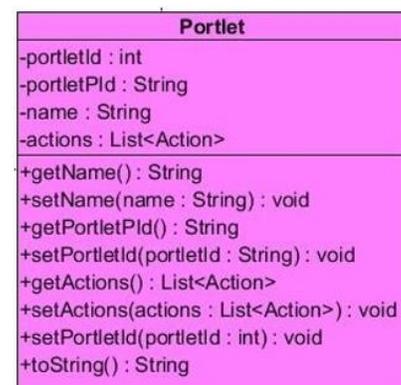
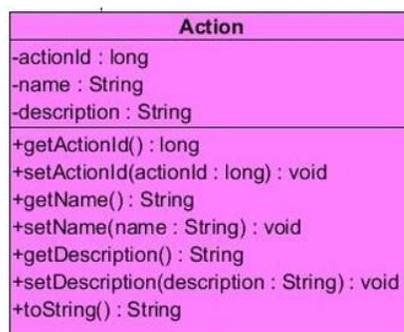
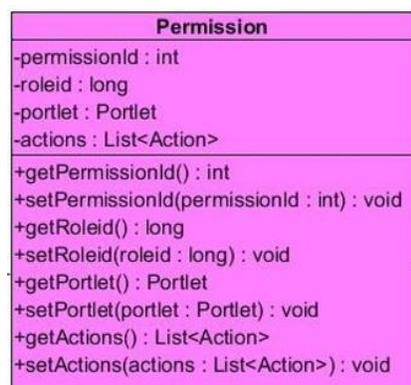
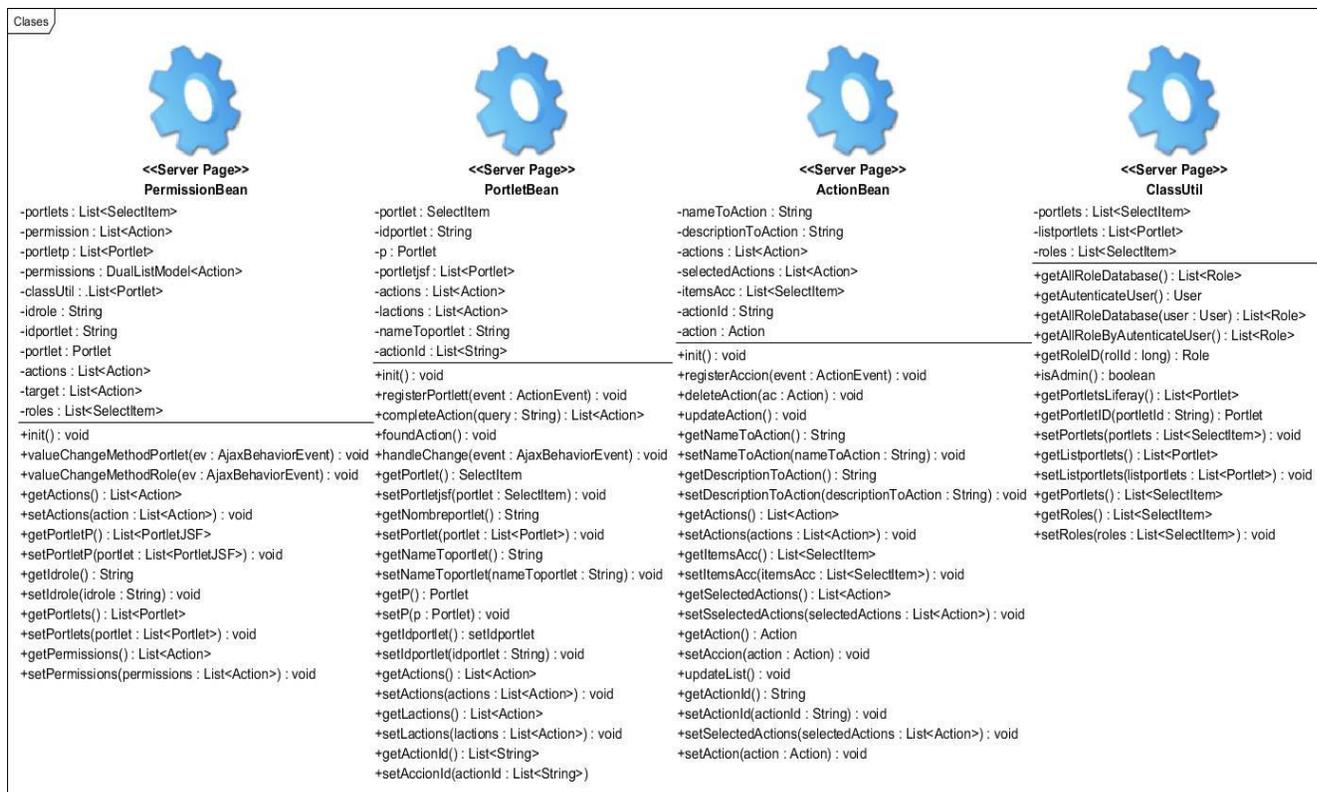


**Anexo 14 Tendencia de búsqueda: Librerías de componentes visuales para JSF.**

- PrimeFaces
- RichFaces
- IceFaces

Orden	Producto	1. Content Management	2. Collaboration	3. Social	4. Security	5. Integration Platform	6. Extensibility and easy Development	7. Ready for Mobility	8. Community (user forums, documentation, wiki, etc.)	9. License type (Free Open Source, Commercial Open Source, Proprietary)	Puntos	Posición
1	a. Alfresco ECM	5	5	3	4	5	5	0	4	Dual	31	3
2	b. Apache Pluto	1	0	0	2	3	3	0	2	FOS	11	14
3	c. Apache Jetspeed-2	3	1	0	3	4	4	0	2	FOS	17	11
4	d. eXo Platform	5	5	4	4	5	5	3	3	Dual	34	2
5	e. Hippo Portal 2	4	2	0	3	4	4	0	2	FOS	19	9
6	f. IBM WebSphere Portal	3	4	4	4	5	5	2	3	PTRY	30	4
7	g. InfoGlue	2	2	0	3	3	3	0	2	FOS	15	12
8	h. Jahia	4	3	2	3	3	4	3	2	Dual	24	8
9	i. JBoss GateIn	3	1	1	3	4	4	0	2	Dual	18	10
10	j. Liferay Portal	4	5	4	5	5	5	2	4	Dual	34	1
11	k. OpenPortal	0	2	0	3	3	3	0	1	FOS	12	13
12	l. Oracle Portal	3	5	4	4	3	5	2	2	PTRY	28	6
13	m. Sakai	4	4	3	4	4	4	2	3	FOS	28	5
14	n. uPortal	4	4	3	4	4	4	1	4	FOS	28	7

**Anexo 15 Portales Corporativos Web 2.0: Soluciones existentes (2015).**



**Anexo 16 Clase con atributos y funcionalidades.**