



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

CENTRO DE CONSULTORÍA Y DESARROLLO DE ARQUITECTURAS EMPRESARIALES

Componente para gestión de trazas del Liferay Portal

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

Autor: Brian Valdés Martínez

Tutores: Dr. Orestes Febles Díaz

Ing. Susana Alba Silot

La Habana, enero de 2016

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Orestes Febles Díaz
Tutor

Susana Alba Silot
Tutora

Brian Valdés Martínez
Autor

Tutor: Orestes Febles Días.

Ciudadanía: Cubana.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Doctor en Ciencias.

E-mail: ofebles@uci.cu.

Tutora: Susana Alba Silot.

Ciudadanía: Cubana.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniera en Ciencias Informáticas.

E-mail: salba@uci.cu.

AGRADECIMIENTOS

Yo agradezco primeramente a mis padres que han dado todo el esfuerzo para que yo ahora este culminando esta gran etapa de mi vida y darle gracias por apoyarme en todos los momentos difíciles de mi vida: Madelaine, René, María, Elman, Lidia y El Berna.

Agradezco a mi novia Lidia María que siempre está a mi lado en todas mis decisiones y apoyarme para hacerlas realidad.

Agradezco a todos mis familiares en los cuales me apoye siempre en las buenas y malas para salir adelante en esta vida, a mi tía Maura, a mi hermano Andy, Petunia, Camilo y mi tío Pedrín. Agradezco a mis tutores Orestes Febles Días y Susana Alba Silot por confiar en mí además del apoyo que me brindaron en cada momento.

Agradezco a las personas que en un momento crucial me ayudaron al desarrollo de esta investigación, Leduan Bárbaro Rosell, Yurien Santos Valdés, Armando Masó Mosqueda, Osmel Pérez Azola, Alain Ramos Medina y Eliuvis Matos Matos.

Agradezco a los amigos que he fomentado en mi paso por la UCI, Raúl Pérez Guevara, Leduan Bárbaro Rosell, Alik Ramón del Risco, Dashiel Lázaro Portel Batista, Oscar Romero Silva, Papo, y a otros muchos más con los que compartí momentos inolvidables.

Agradezco a todos los profesores que me mostraron el camino para lograr la obtención de este título como Yoan Parra, Olga Fontova, las profes de inglés Zaida, Abi y María Cristina, entre otros.

A Rapuncell y Ragnar por darme tanto trabajo y cariño en este último período de la carrera.

DEDICATORIA

A mis padres, a mi abuela Toña y mi novia Lidia por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.

Todo este trabajo ha sido posible gracias a ellos.

RESUMEN

En la actualidad las empresas se ven con la necesidad de alcanzar diferencias competitivas, lograr agilidad en el proceso de negocio y generar servicios a través de la utilización de las nuevas tecnologías. Necesidades que estarán satisfechas con el empleo de disímiles herramientas, sobresaliendo el uso de los portales empresariales, encaminados para brindarles a los usuarios, de manera concreta y fácil, el acceso a servicios concernientes de un tema. Incluyendo buscadores, documentos, foros, aplicaciones, compra electrónica, enlaces, entre otros. Liferay Portal *Community Edition* (CE) es una plataforma que tiene como principio entregar el máximo de beneficios a sus usuarios. Cuenta con más de 60 aplicaciones *portlet* (predefinidas), listas para desplegar las funciones básicas para que un proveedor de servicios realice una interrelación fácil, eficiente y segura con sus usuarios. La presente investigación se nutre de la generación de un componente de gestión de trazas, el cual brindará fiabilidad en el uso de la herramienta y apoyo a los administradores del sistema en la toma de decisiones. El proceso de desarrollo del componente fue manejado a través de la Metodología Unificada Ágil (AUP) con una pequeña adaptación para los procesos productivos de la Universidad de la Ciencias Informáticas (UCI). Para la conformación de la aplicación se emplearon tecnologías y herramientas libres (*Open sources*), a excepción de Visual *Paradigm for UML*. Para comprobar el correcto funcionamiento del componente se realizaron pruebas de caja negra, de aceptación y rendimiento arrojando resultados positivos.

Palabras clave: auditoría, componente de gestión de trazas, registro.

CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1 Actualidad en el desarrollo de portales empresariales y sus principales componentes	6
1.1 Web 2.0	6
1.1.2 Sistema gestor de contenidos (CMS)	7
1.2 Portales Web	8
1.3 Liferay Portal.....	9
1.3.1 Portlet.....	11
1.3.2 Herramientas asociadas al desarrollo sobre Liferay.	11
1.4 Seguridad informática y Auditoría	20
1.5 Conclusiones	22
CAPÍTULO 2 Componente para la gestión de trazas y eventos del Liferay Portal.....	23
2.1 Propuesta de solución	23
2.2 Ingeniería del Software basada en componentes (ISBC).....	23
2.3 Service Builder.....	25
2.4 Estructura del componente	27
2.5 Requisitos del sistema	27
2.5.1 Requisitos funcionales (RF).....	28
2.5.2 Requisitos no funcionales (RNF)	29
2.6 Historias de Usuario.....	29
2.7 Modelo de domino	35
2.8 Estilo arquitectónico y patrones utilizados.....	36
2.9 Conclusiones parciales	38
CAPÍTULO 3 Implementación y Pruebas	39

3.1	Estándar de código	39
3.2	Pruebas	40
3.2.1	Estrategia de pruebas	41
3.2.2	Casos de pruebas	43
3.2.3	Lista de chequeo	45
3.3	Pruebas de aceptación	46
3.4	Pruebas de estrés.....	50
3.5	Conclusiones parciales	51
	CONCLUSIONES	52
	RECOMENDACIONES	53
	BIBLIOGRAFÍA (referencias)	54
	ANEXOS.....	58

Índice de ilustraciones:

Ilustración 1: Web 2.0 (PROFESORADO, 2012).....	6
Ilustración 2: Liferay Portal Community Edition (LIFERAY, 2014a).	10
Ilustración 3: Service Builder (RICHARD SEVOZ, 2012).....	12
Ilustración 4: Archivo service.xml. Elaboración propia.....	26
Ilustración 5: Estructuras de los paquetes generados por el service builder. Elaboración propia.	26
Ilustración 6: Modelo de dominio. Elaboración propia.	35
Ilustración 7: Arquitectura en 3 Capas. Elaboración propia.	36
Ilustración 8: Patrón Modelo Vista Controlador. Elaboración propia.	37
Ilustración 9: Interfaz principal del componente. Elaboración propia.	40
Ilustración 10: Prueba para 50 usuarios utilizando jmeter 2.3.1. Elaboración propia.	51
Ilustración 11: Cuadrante mágico de Gartner (GARTNER, 2015).....	58
Ilustración 12: Interfaz de filtrado por la descripción del evento. Elaboración propia.	58
Ilustración 13: Ventana emergente de descarga del documento a formato excel. Elaboración propia.....	59
Ilustración 14: Reporte a formato excel. Elaboración propia.....	59
Ilustración 15: Ventana emergente de descarga del documento a formato pdf. Elaboración propia.	60
Ilustración 16: Reporte a formato pdf. Elaboración propia.	60
Ilustración 17: Interfaz del tablero de control. Elaboración propia.....	61
Ilustración 18: Gráfico de conexiones por dominio. Elaboración propia.....	61
Ilustración 19: Gráfico de páginas más visitadas. Elaboración propia.	62
Ilustración 20: Gráfico de conexiones diarias. Elaboración propia.....	62
Ilustración 21: Gráfico de conexiones en los últimos 7 días. Elaboración propia.	63
Ilustración 22: Interfaz de filtrado por fecha. Elaboración propia.	63
Ilustración 23: Interfaz de filtrado por hora. Elaboración propia.	64
Ilustración 24: El componente Audit Portlet del Liferay Portal (EE) en el panel de control (LIFERAY, 2014a).....	64
Ilustración 25: El componente Event Trace Portlet del Liferay Portal (CE) en el panel de control. Elaboración propia.	65

Ilustración 26: Estructura del proyecto generado con maven. Elaboración propia.....65

Ilustración 27:Interfaz del componente Audit Portlet del Liferay Portal (EE) (LIFERAY, 2014a).
.....66

Índice de tablas:

Tabla 1: Tabla de Requisitos funcionales.....	28
Tabla 2: Tabla de Requisitos no funcionales.....	29
Tabla 3: HU Registrar eventos de los usuarios.	30
Tabla 4: HU Obtener listado de los eventos de los usuarios.	31
Tabla 5: HU Filtrado por campos.....	32
Tabla 6: HU Reporte (exportar) a formato pdf o Excel.....	33
Tabla 7: HU Visualizar el comportamiento de las conexiones.	34
Tabla 8: HU Tablero de control para visualizar el comportamiento de los temas analizados.	34
Tabla 9: Descripción del modelo conceptual.	35
Tabla 10: Estrategia de pruebas.	43
Tabla 11: Caso de prueba EC 3: Filtrado por campos.....	45
Tabla 12: Lista de chequeo.	46
Tabla 13: Pruebas de aceptación [RF1.].....	47
Tabla 14: Pruebas de aceptación [RF2.].....	48
Tabla 15: Pruebas de aceptación [RF3.].....	48
Tabla 16: Pruebas de aceptación [RF4.].....	49
Tabla 17: Pruebas de aceptación [RF4.] 2da iteración.	49
Tabla 18: Pruebas de aceptación [RF5.].....	50
Tabla 19: Pruebas de aceptación [RF6.].....	50
Tabla 20: Escenarios de prueba (continuación).	69

INTRODUCCIÓN

Hoy en día la web es un universo de aplicaciones y páginas web interconectadas y cargadas de fotos, videos, información y contenido interactivo. Con el surgimiento de la web 2.0 se pasa de una web de información creada por expertos a una web social donde cualquier usuario puede participar fácilmente.

Aparecen portales web muy potentes y sencillos de manejar enfocados al usuario final. En ellos se les ofrece a los internautas de forma fácil e integrada el acceso a una serie de recursos y servicios relacionados con un tema en específico, incentivando en los usuarios cada vez la dependencia hacia los mismos.

El creciente uso de la informática en todas las esferas del desarrollo científico, económico, político y social, así como el surgimiento de nuevos riesgos asociados principalmente con el uso de las redes de datos de alcance global, se hace necesario el desarrollo de sistemas que sean capaces de regular la seguridad informática, disciplina que se relaciona a diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información de un sistema informático y sus usuarios.

Con el paso de los años las grandes empresas han señalado la necesidad de realizar un estudio detallado del comportamiento de los procesos ejecutados en los grandes sistemas al hacer uso de portales corporativos basados en la web, para ello la manera de realizar esta observación sería a través del uso frecuente de las auditorías. Estas llevan a cabo un control acerca del uso correcto de los recursos que brindan las instituciones, la tendencia de los usuarios, detectar sus vulnerabilidades, prevenir errores y fraudes, así como emitir una opinión acerca de la marcha de la empresa, evaluar sus metas, además de efectuar un posterior seguimiento de las recomendaciones (GALVIS *et al.*, 2009). Es a través de este proceso de evaluación que se obtiene competitividad en el mercado, se logra una mayor interrelación con el cliente y se hace uso eficiente de la tecnología.

Disponer de un buen *software* corporativo es primordial no solo para lograr aumentar la productividad de la propia empresa, sino también para medirla y saber qué puede ir mejor o qué puede estar fallando. Teniendo como objetivo más importante el descenso del caos que existe en muchas compañías en la manipulación, el almacenamiento, la facilitación, la transferencia y publicación de conocimientos e informaciones.

En la actualidad múltiples organizaciones utilizan portales empresariales para gestionar las relaciones con sus usuarios. La retroalimentación de estos usuarios es un factor clave para mejorar las plataformas que facilitan el desarrollo de portales. Existen varias herramientas que facilitan el desarrollo de un sitio web hoy en día, entre los que destacan WordPress, Joomla,

Drupal, OpenCMS y Liferay Portal. Estos se encuentran en la categoría de Sistemas de Gestión de Contenidos (por sus siglas en inglés CMS). Son soluciones ampliamente utilizadas dentro y fuera del mundo empresarial, existiendo un incansable entusiasmo y competitividad entre ellas que las hace evolucionar constantemente. Los CMS permiten a un editor gestionar los contenidos de una página web, crear, clasificar y publicar cualquier tipo de información, sin embargo, los portales empresariales brindan además la gestión de los contenidos de una página web, servicios integrales como son los buscadores, foros, wiki entre otros y están dirigidos a un grupo de personas.

Uno de los portales empresariales que goza de prestigio y aceptación además de incluirse en la categoría de CMS es Liferay Portal, el cual permite gestionar el contenido y apariencia web a través de un modelo de componentes, mediante una serie de interfaces visuales como si fuera el típico escritorio con sus ventanas, moverlas de sitio, añadir, eliminar, entre otras. Este es el punto más característico de esta herramienta respecto a los CMS convencionales (ROSAS, 2010).

Además, es compatible con los servidores de aplicaciones líderes y bases de datos, como PostgreSQL Server y MySQL, es un sistema multiplataforma compatible con Linux, UNIX y Windows. Al mismo tiempo es una de las plataformas web corporativa *Open source* más utilizada en el mundo, ubicada en el cuadrante de portales horizontales líderes según la empresa consultora y de investigación de las tecnologías de la información Gartner (GARTNER, 2015).

Otra fortaleza de la herramienta es que cuenta con más de 60 aplicaciones (*portlets*) listas para utilizar. Un *portlet* es un componente modular (pequeña aplicación web) visualizada en un portal web (CANO, 2007). Permite persistir las preferencias en base de datos y una vez reiniciado el sistema no pierde dichas configuraciones.

Este portal es la base para compañías de empresas prestigiosas como son:

- Allianz (Portal del grupo asegurador Allianz en España).
- Basf (una de las empresas de la industria química más importantes del mundo).
- Cisco (líder mundial en soluciones de redes).
- Ministerio de Defensa Francés o Naciones Unidas.

Liferay les brinda a sus clientes la posibilidad de contar con dos ediciones:(LIFERAY, 2012)

- Liferay Portal *Enterprise Edition* (EE), la cual cuenta con una herramienta de desarrollo Liferay *Developer Studio*, incluye 100 características de flujo de trabajo, formas web, integración de sistemas, además de requerir una licencia comercial.
- Liferay Portal *Community Edition* (CE), cuenta con una herramienta de desarrollo Eclipse IDE plugin, más de 60 aplicaciones *portlets* y cuenta con una licencia *Open source*.

Nuestro país en aras de lograr un mayor avance tecnológico en los procesos de negocios ha hecho uso del Liferay Portal *Community Edition* (CE) para agilizar el proceso de desarrollo de portales como:

- Sistema informativo del Ministerio de la Colaboración Económica en el Extranjero (SI-MINCEX).
- Sistema para la Gestión de operaciones del Centro de Inmunología Molecular (CIM).
- Proyecto de Integración para el Ministerio de interior y justicia de la República Bolivariana de Venezuela (MPPRIJP).

Todos pertenecientes al Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), de la Universidad de Ciencias Informáticas (UCI). Esta plataforma web es potente, segura y flexible con la cual se desarrollaron con gran calidad los portales antes mencionados, sin embargo, un grupo de administradores identificaron una serie de insuficiencias relacionadas a la gestión de trazas con las siguientes características:

- Permitir monitorear, estudiar y evaluar el comportamiento de los usuarios dentro del sistema.
- Almacenar los procesos de autenticación, autorización y salida de los usuarios al sistema, ya que el Liferay Portal *Community Edition* no registra esos eventos.
- Existe un componente denominado *Audit Portlet*, desarrollado para la versión Liferay Portal (EE) el cual es privado y para su obtención es necesario pagar por el producto.
- Al no tener un mecanismo que permita la gestión de las trazas la herramienta pierde la capacidad de ser auditable.

Para los administradores y desarrolladores de portales basados en Liferay Portal (CE), contar con un componente de gestión de trazas les permite fiabilidad en el uso de esta plataforma.

Partiendo de la problemática se elabora como **problema de investigación**:

¿Cómo gestionar eficientemente las trazas de los eventos realizados en Liferay Portal (CE) para lograr fiabilidad en el uso de esta herramienta y apoyar la toma de decisiones de los administradores del portal?

Se enmarca como **objeto de estudio de la investigación**:

La gestión de las trazas y eventos en los portales web.

Como **campo de acción**:

La gestión de trazas y eventos en Liferay Portal.

Para resolver la problemática se plantea como **Objetivo General**:

Desarrollar un componente reutilizable para gestionar las trazas de los eventos lanzados por Liferay Portal.

Objetivos específicos a realizar para lograr el objetivo formulado:

- Elaboración del estado del arte sobre las tendencias actuales de los portales, así como los principales conceptos que se emplean durante la investigación.
- 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.
- Elaboración e implementación del componente de gestión de trazas y eventos.
- Realizar pruebas al componente que validen el cumplimiento de los objetivos de la investigación.

Se emplearon métodos varios para el perfeccionamiento de las tareas investigativas como:

Métodos teóricos empleados:

El **Análisis-Síntesis**, método que se utilizó para lograr una mayor profundidad acerca del funcionamiento del Liferay Portal, arquitectura que presenta, características fundamentales y elaborar conclusiones mediante la síntesis de los conocimientos y resultados obtenidos.

La **Modelación**, método lógico que consiste en describir a través de diagramas, esquemas, o representaciones la estructura de relaciones y propiedades fundamentales del *portlet*.

Métodos empíricos definidos:

La **Observación Científica**, permite valorar las diversas bibliografías referentes a las metodologías, técnicas y herramientas a utilizar para dar solución a la problemática planteada.

El método de **Consulta de la Información** en todo tipo de fuente, es manejado para confeccionar el marco teórico o el estado del arte de la indagación, permitiendo acceso a bibliografías de diversas opiniones las cuales han sido referenciadas para percibir mejor la situación problemática, y decidir si es posible llevarla a cabo tanto científica como económicamente.

Este documento científico estará compuesto por 3 capítulos:

Capítulo I. Actualidad en el desarrollo de portales empresariales y sus principales componentes. Capítulo donde se estudian las tendencias actuales en el desarrollo de los *portlets* realizados para el Liferay Portal. Se define el lenguaje de programación, metodología, tecnologías y herramientas para la implementación de la aplicación que dará solución al problema planteado.

Capítulo II. Componente para la gestión de trazas y eventos del Liferay Portal. Componente de Gestión de trazas y eventos para Liferay Portal (CE) como propuesta de solución que permitirá obtener la información de los eventos realizados por los usuarios dentro del sistema: tomándose como base la metodología AUP para orientar el proceso de desarrollo.

- **Capítulo III. Implementación y pruebas.** Análisis y diseño del *portlet* de gestión de trazas y eventos para portales desarrollados por el Liferay Portal (CE): Se expone lo concerniente al flujo de implementación y se brinda una solución a las insuficiencias encontradas. Se diseñan, describen y realizan los casos de pruebas aplicados al sistema para probar la eficiencia del mismo.

1.1.2 Sistema gestor de contenidos (CMS)

Software que permite la creación y administración de los contenidos de una página Web, principalmente, de forma automática. Así, con él podemos publicar, editar, borrar, otorgar permisos de acceso o establecer los módulos visibles para el visitante final de la página. El CMS está formado por 2 elementos: (MAYOR, 2014)

- La aplicación gestora de contenidos (CMA): El elemento CMA permite al gestor de contenidos o autor realizar la creación, modificación y eliminación de contenido en un sitio Web sin necesidad de tener conocimientos de lenguaje HTML.
- La aplicación dispensadora de contenidos (CDA): El CDA usa y compila la información para actualizar el sitio Web.

Cuando hablamos de un CMS nos referimos a una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. Las características de un CMS pueden variar, pero la mayoría incluye publicación basada en Web, indexación, revisión, búsqueda y recuperación de la información (MAYOR, 2014).

Ejemplos de CMS

WordPress es una plataforma donde el usuario puede escribir, modificar artículos y crear una página web o un blog. Famoso por su capacidad de ser rápido, sencillo y fácil de aprender a usar. Este permite publicar todo tipo de información sin tener que preocuparse por introducir código, además cuenta con un amplio grupo de plantillas WordPress o temas WordPress los cuales pueden ser modificados si el usuario lo desea. Un rasgo característico de este CMS es que es gratis, no es necesario un pago adicional por utilizarlo o por publicar blogs o sitios web en Internet. WordPress frecuentemente está sujeto a cambios (actualizado y modernizado) por mediación de las opiniones y sugerencias de usuarios.

Joomla es un gestor de contenidos para el desarrollo web: es una especie de programa para gestionar páginas web. Se basa en estándares propios de desarrollos web como XHTML, CSS, *JavaScript* y PHP. Es una aplicación web, es decir, no se ejecuta en el equipo local sino en un servidor (aunque se puede hacer funcionar en el equipo local como servidor). Joomla es *software* libre: está desarrollado por una comunidad de usuarios que lo mejoran, actualizan y lo ponen a disposición para su uso gratuitamente. Permite publicar sitios web cuyos contenidos cambian continuamente, fáciles de administrar y ampliar sin necesidad de realizar una inversión económica en *software*. Continúa ampliando a través de nuevos módulos y plantillas que se van publicando periódicamente. Joomla permite tanto la publicación por parte del gestor de la página web como la interacción de los usuarios de la web a través de publicación de artículos, encuestas o colaboraciones en la gestión del sitio. Además, es una solución indicada para

personas que quieren publicar una web centrándose en los contenidos y dejando que el *backup* se base en un desarrollo de *software* libre (RANCEL, 2009).

Drupal es un CMS o sistema de gestión de contenidos que se utiliza para crear sitios web dinámicos y con gran variedad de funcionalidades. Drupal es un *software* libre, escrito en PHP, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores que colaboran conjuntamente en su mejora y ampliación. Esta ampliación es posible gracias a que se trata de un sistema modular con una arquitectura muy consistente, que permite que los módulos creados por cualquier desarrollador puedan interactuar con el núcleo del sistema y con los módulos creados por otros miembros de la comunidad. Con Drupal es posible implementar una gran variedad de sitios web: un blog personal o profesional, un portal corporativo, una tienda virtual, una red social o comunidad virtual (BUITRAGO, 2011).

OpenCms es un sistema de gestión de contenido de fuentes abiertas basado en Java y en tecnología XML. Es distribuido por la empresa Alkacon *Software* bajo licencia LGPL (TECNOLÓGICAS, 2010).

Se trata de una aplicación CMS con características tales como entorno de trabajo basado en navegador web, gestión de activos, sistemas de gestión de usuarios y permisos integrados, publicación de contenidos basada en proyectos, gestión de tareas, soporte a la internacionalización, versionado del contenido, mecanismos de plantillas JSP y XML, soporte Multi-idioma, sistema de ayuda *online*, publicación dinámica y estática de contenidos, personalización, sistemas de cacheo integrados, mecanismo modular para las extensiones, sistema de programación de trabajos, mecanismo de sincronización, importación y exportación de contenidos, integración con el servidor de aplicaciones y soporte para *Enterprise Java Bean* (EJB) (TECNOLÓGICAS, 2010).

1.2 Portales Web

Tratados de puntos o puertas de entrada a Internet, donde a través de los cuáles se le ofrece al usuario el acceso directo a un conjunto heterogéneo de servicios desde la página web de entrada.

Se pueden distinguir 2 tipos de portales:

- Portales generalistas: ofrecen al usuario un conjunto de servicios relacionados con Internet tales como un buscador por palabras clave, directorio temático, correo electrónico gratuito, acceso a internet, chats, listas de distribución, mensajes a teléfonos móviles, etc. y otra serie de servicios de interés general como agenda, callejero, noticias, compras en línea, distintos canales temáticos, etc. Cuando los portales dan a los usuarios la posibilidad de que cuelguen sus propias páginas web dentro del portal, suelen constituirse comunidades virtuales de usuarios unidos por los temas que tienen en

común. La diferencia entre un portal y un buscador es que el primero ofrece, además de los servicios de búsqueda, otra serie de servicios sin que el usuario tenga que buscarlos por sí mismo (LAPUENTE, 2013).

- Portales especializados: brindan acceso a servicios sobre un tema concreto. Denominados también portales verticales, tienen su propia identidad y estructura de contenidos, desplegados con el fin de facilitar a los internautas el acceso a un volumen considerable de información organizada para temas específicos (LAPUENTE, 2013).

Portal Empresarial

El portal empresarial es una alternativa de la tecnología para que las empresas integren en los sitios de Internet la colaboración con sus clientes, aliados de negocios, proveedores, y otros, utilizando una infraestructura segura para el intercambio de información.

Uno de los grandes cambios que las organizaciones están buscando es aumentar la competitividad, a través de medios para conectar a las personas, la información y los procesos del negocio para un mejor posicionamiento en el mercado y eliminar ineficiencias (INSPIRAIT, 2015).

Clasificaciones de los portales atendiendo a los servicios a los que se podrán acceder: (LEBRÚN, 2005)

Portales de Negocios

Los portales de Negocios son aquellos que las empresas ponen a disposición de los clientes y prospectos, y que vienen a ofrecer una serie de servicios ya clásicos, que pueden ir desde servicios de correo electrónico, agendas personalizadas o comercio electrónico, hasta servicios de atención al cliente, consulta de datos de facturación, o acceso a aplicaciones de todo tipo. Lo normal es que estos servicios sean acompañados con una serie de contenidos, pudiendo ser contenidos generales o centrados en un sector o área de actividad.

1.3 Liferay Portal

El **Liferay Portal** fue creado en el año 2000 es una plataforma web empresarial *Open source* para construir aplicaciones de negocios. Fue diseñado para facilitar el acceso a las distintas aplicaciones o contenido de información institucional que se desee administrar bajo un entorno de trabajo unificado. Liferay es utilizado a nivel mundial para sitios web públicos con funcionalidades de administrador de contenidos (CMS), intranets, internet, plataformas de colaboración y aplicaciones para la web. Liferay es una alternativa real a portales como Websphere Portal, Oracle Portal, JBoss Portal o Jetspeed (ROSAS, 2010). Además, es considerado líder a nivel mundial entre los portales web de gestión de contenidos de código abierto.

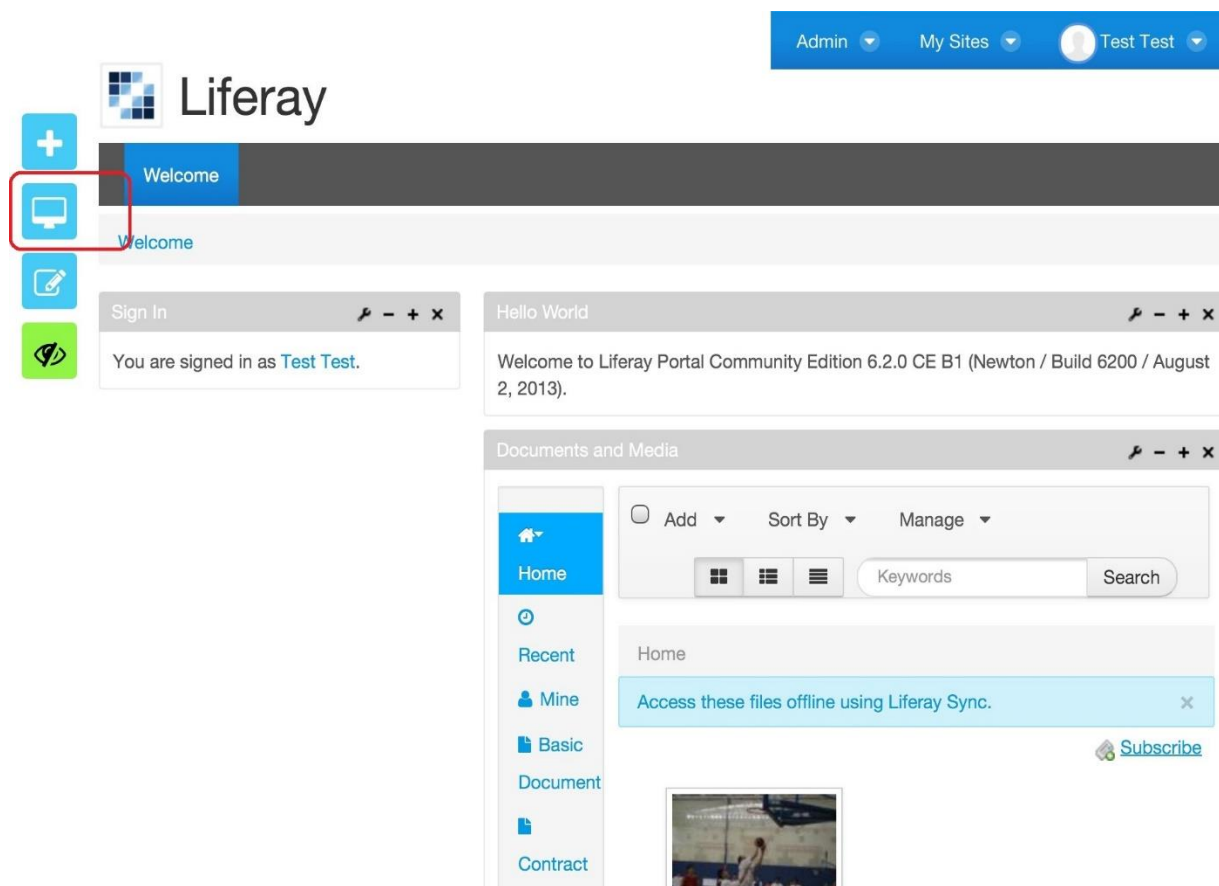


Ilustración 2: Liferay Portal Community Edition (LIFERAY, 2014a).

Fundamentos de Liferay

Las funcionalidades de Liferay podrían agruparse en dos conjuntos. Por un lado, la configuración estándar de Liferay permite administrar las funcionalidades más básicas en cuanto a usuarios, grupos de usuarios, roles, organizaciones y comunidades. Estas funcionalidades, aunque a simple vista aparenten ser simples, hacen posible la personalización del portal de acuerdo al usuario, llegando incluso a considerar roles con privilegios sumamente específicos (ROSAS, 2010).

Por otra parte, Liferay provee un conjunto predefinido de herramientas denominado Liferay *Collaboration Suite*, un juego de aplicaciones que pueden utilizarse para construir comunidades de usuarios que utilicen intensivamente el portal. Esta suite contiene blogs, calendarios, mensajería instantánea (chat), correo electrónico (para servidores ya existentes y que soporten IMAP), tablero de anuncios (foros) y wikis (ROSAS, 2010).

1.3.1 *Portlet*

Los *portlets* hacen referencia a una pequeña aplicación del portal, usualmente diagramados como un componente dentro de la página web. Un *portlet* puede ser desarrollado, publicado y ejecutado independientemente de otro *portlet*, existiendo una serie de *portlets* ya disponibles para las funcionalidades básicas del portal (e-mail, calendario, chat, etc.), los cuales sólo es preciso configurarlos para su implementación (ING. JAVIER A. VOOS, 2012).

Los *portlets* no son sólo simples vistas de contenido web, son aplicaciones completas, que resuelven una problemática determinada, para ello cuentan con sus propios datos y pantallas, con la posibilidad de integrarse a otros *portlets* ya publicados en el portal. Cuentan con un entorno de ejecución (*portlet container*) provisto por la infraestructura del portal, que es utilizada además para acceder a otros servicios como seguridad, manejo de ventanas, personalización, acceso a contenido remoto, entre otras. Este entorno de ejecución está basado en una Interfaz de Programación de Aplicaciones (API) que provee las clases e interfaces necesarias para programar los *portlets*, lo que posibilita abstraerse de los mecanismos utilizados a bajo nivel por el portal para su ejecución (ING. JAVIER A. VOOS, 2012).

Principales características:

- Los *portlets* son exclusivamente generados como parte de una página web (no son documentos completos), por lo que no son asociados a una URL específica.
- Los *portlets* son controlados por un contenedor especializado, el cual controla su ciclo de vida.
- Los *portlets* a través del uso del paradigma *request/response* (solicitud/respuesta), interactúan con el cliente, generando contenido dinámico para dar solución a la funcionalidad solicitada.
- Los *portlets* proveen un objeto *Portlet Preferences* para conservar las preferencias del usuario.

1.3.2 **Herramientas asociadas al desarrollo sobre Liferay.**

Para la confección del componente el autor de esta investigación selecciona herramientas que sean fáciles de integrar, tengan soporte y contengan abundante información para el uso eficiente de estas y lograr la calidad esperada del componente.

Hooks

Los *hooks* se utilizan para desarrollar *plugins* que sobrescriben partes del 'core' de Liferay, pero sin tener que tocarlo (modificarlo), ni usar el entorno de extensión EXT. Se trata de una forma elegante de modificar el comportamiento de Liferay, permitiendo mantener una buena gestión de esas modificaciones, evita en un futuro tener complicaciones de compatibilidad o mal

funcionamiento de los componentes que intervienen en la generación del portal (CORONADO, 2011).

Tipos de *Hooks* para el Liferay: (CORONADO, 2011)

- Sobrecribir ficheros JSP.
- Sobrecribir servicios del portal.
- Sobrecribir las traducciones.
- Sobrecribir *Model Listeners*.
- Insertar eventos en el ciclo de vida de Liferay.

Service Builder

El *Service Builder* es una herramienta para Liferay, presente en el IDE, que nos permite desarrollar rápidamente aplicaciones basadas en datos. En la práctica, a partir de la definición de una entidad en formato xml se puede generar automáticamente todas las clases en la gestión de *back-end* de la persistencia de una aplicación. Todo esto se basa en el uso de Spring e Hibernate, pero la ventaja es que la herramienta realiza la generación de distintos archivos de configuración, las clases DAO y servicios que de otro modo se haría de forma manual. Todo esto es fácilmente extensible y personalizable (SCAMARIO, 2010).

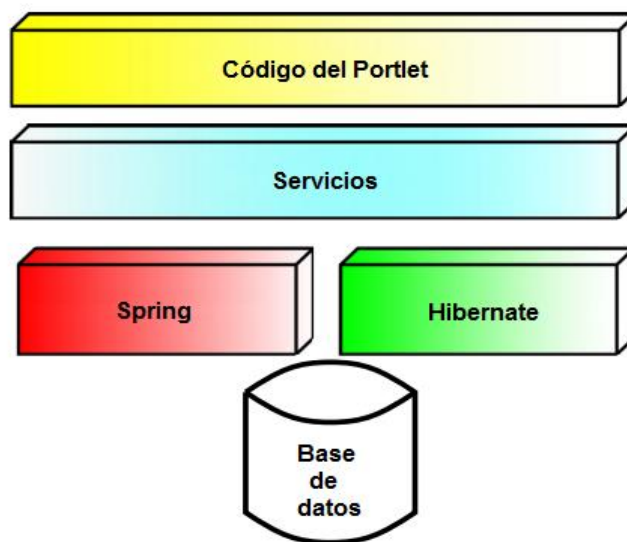


Ilustración 3: *Service Builder* (RICHARD SEVOZ, 2012).

Maven

Maven, herramienta *open-source*, creada en el 2001, cuyo objetivo es simplificar los procesos de *build* (compilar y generar ejecutables a partir del código fuente). Capaz de ocuparse de la gestión de un proyecto completo, desde la etapa en la que se comprueba que el código es el correcto, hasta que se despliega la aplicación. Maven posee un archivo, Project Object Model (POM), encargado de la gestión de dependencias entre los módulos y las diferentes versiones de las bibliotecas, que de otro modo sería a mano. Esta herramienta maneja 3 niveles de repositorios, local, empresarial y global, cuando comienza el proceso de compilación, maven busca las dependencias del proyecto en el repositorio local, si no las encuentra, intenta encontrarlas en los repositorios empresariales definidos en el settings.xml (archivo), y en caso de no encontrar los jar (archivo java) necesarios, los busca en los repositorios globales. Una vez ubicado el jar en los repositorios externos, maven procede a descargarlo hacia su repositorio local para su correcto funcionamiento.

Java (Lenguaje de programación)

El lenguaje de programación orientado a objeto Java, creado por Sun Microsystems en el año 1991, brinda la posibilidad de ser un lenguaje que puede ser ejecutado independientemente del contexto, ambiente; llegando a ser la portabilidad uno de sus principales logros. Java es un lenguaje de código abierto por lo que puede ser modificado si el usuario lo desea, además su uso no implica inversiones económicas, no es necesario adquirir licencia alguna, solo descargar el *kit* de desarrollo (Java Development Kit o JDK). Java cuenta con una máquina virtual (conocida como Java Virtual Machine o JVM) la cual ejecuta las aplicaciones desarrolladas independientemente del sistema, por lo que funciona en cualquier entorno (multiplataforma) Linux, Windows, Mac OS y para cualquier sistema en el cual exista una JVM.

Se escoge el lenguaje de programación Java para generar la solución al problema antes planteado, por ser el Liferay Portal un CMS de código abierto e implementado en Java.

Framework Spring

Spring es un *framework* liviano y no intrusivo: generalmente los objetos que se programan no tienen dependencias en clases específicas de Spring. Sus características principales son **inyección de dependencias y programación orientada a aspectos** (BRIANO, 2010).

Módulos de Spring: (BRIANO, 2010)

Core – se encuentra el **BeanFactory** – el contenedor fundamental de Spring y quien se encarga de la inyección de dependencias. El contenedor **ApplicationContext** se basa en **BeanFactory** y extiende la funcionalidad con soporte para i18n, eventos de ciclo de vida, validación y mejor integración con AOP.

AOP (Programación orientada a aspectos) – provee la implementación de AOP, permitiendo desarrollar interceptores de método y puntos de corte para desacoplar el código de las funcionalidades transversales.

DAO (Objeto de acceso a datos) – Provee una capa de abstracción sobre JDBC, abstrae el código de acceso a datos de una manera simple y limpia. Tiene una capa de excepciones sobre los mensajes de error provistos por cada servidor específico de base de datos. Además, cuenta con manejo de transacciones a través de AOP.

ORM (Mapeo de Objeto Relacional) – Provee la integración para las distintas APIs de mapeo objeto-relacional incluyendo JPA, JDO, Hibernate e iBatis.

JEE (Java Enterprise Edition) – Provee integración con aplicaciones Java *Enterprise Edition* así como servicios Java *Management Extension (JMX)*, Java *Management Service (JMS)* y *Enterprise Java Beans (EJB)*.

Web – Módulo que aporta clases especiales orientadas al desarrollo web e integración con tecnologías como *Struts* y *JSF*. Cuenta con el paquete **Spring MVC (Modelo vista - controlador)**, una implementación del conocido patrón de diseño aplicando los principios de Spring.

Spring Framework es un marco de aplicaciones Java que facilita la implementación de diferentes patrones de diseño y la composición con disímiles tecnologías.

Framework Hibernate

Hibernate es un potente servicio de persistencia Objeto – Relacional de alto rendimiento que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación. Ofrece persistencia automatizada y transparente de objetos a tablas en una base de datos relacional, utilizando metadato que describe el mapeo entre objetos y la base de datos en sí (BRIANO, 2010).

Entre las características más importantes se destacan: (BRIANO, 2010)

- Licencia LGPL.
- Ofrece un lenguaje propio de consulta HQL.
- Implementado con XML o Anotaciones (JPA).
- Excelente Documentación.
- Fácil de aprender.
- Comunidad Activa.
- Soporte para Windows.

Framework JSF

JSF (Java Server Faces) es un marco de trabajo para crear aplicaciones java J2EE basadas en el patrón MVC donde las vistas conocen la acción que se va a invocar en su petición (ALMIRÓN, 2014).

JSF tiene como características principales: (ALMIRÓN, 2014)

- Utiliza páginas JSP para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios HTML.

- Asocia a cada vista con formularios un conjunto de objetos java manejados por el controlador (*managed beans*) que facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.

Introduce una serie de etapas en el procesamiento de la petición, como por ejemplo la de validación, reconstrucción de la vista y recuperación de los valores de los elementos.

- Utiliza un sencillo fichero de configuración para el controlador en formato xml.

- Es extensible, modificando nuevos elementos de la interfaz o modificar los ya existentes.

- Forma parte del estándar J2EE (Java 2 Platform, Enterprise Edition). En efecto, hay muchas alternativas para crear la capa de presentación y control de una aplicación web java, como *Struts* y otros *frameworks*, pero solo JSP (Java Serve Pages) forma parte del estándar.

Porque utilizar JSF para la solución del *portlet*:

JSF 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 (ALMIRÓN, 2014).

JSF ofrece una serie de ventajas: (ALMIRÓN, 2014)

- El código JSF con el que crea las vistas (etiquetas JSP) es muy parecido al HTML estándar. Lo pueden utilizar fácilmente desarrolladores y diseñadores web.
- JSF se integra dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página.
- JSF resuelve validaciones, conversiones, mensajes de error e internacionalización (i18n).
- JSF permite introducir *JavaScript* en la página, para acelerar la respuesta de la interfaz en el cliente (navegador del usuario).
- JSF es extensible, por lo que se pueden desarrollar nuevos componentes a medida. También se puede modificar el comportamiento del *framework* mediante APIs (Interfaz de programación de aplicaciones) que controlan su funcionamiento.

Desde el punto de vista técnico se puede destacar lo siguiente:(ALMIRÓN, 2014)

- JSF forma parte del estándar J2EE, mientras que otras tecnologías para creación de vistas de las aplicaciones no lo forman, como por ejemplo *Struts*.
- JSF dispone de varias implementaciones diferentes, incluyendo un conjunto de etiquetas y APIs estándar que forman el núcleo del *framework*. Entre estas implementaciones cabe destacar la implementación de referencia de Sun Microsystems, actualmente desarrollada como un proyecto *open source*, y la implementación del proyecto Apache, MyFaces, dotada de un conjunto de extensiones que la hacen muy interesante para el desarrollo de aplicaciones corporativas.
- El desarrollo de JSF está realmente empezando. Las nuevas versiones del *framework* recogen la funcionalidad de versiones anteriores siendo su compatibilidad muy alta, de manera que el mantenimiento de aplicaciones no se ve penalizado por el cambio de versiones.

Siendo *Framework* JSF sencillo, robusto y además cuenta con una gran documentación se escoge para la realización de la interfaz visual del componente de gestión de trazas y eventos.

PrimeFaces

PrimeFaces es una librería de componentes visuales *open source* desarrollada y mantenida por *Prime Technology*, una compañía turca de IT especializada en consultoría ágil, JSF, Java EE y Outsourcing. El proyecto es liderado por Çağatay Çivici, un miembro del “JSF Expert Group” (y forofo del Barça).

Las principales características de Primefaces son: (LERMA, 2014)

- Soporte nativo de Ajax, incluyendo *Push/Comet*.
- Kit para crear aplicaciones web para móviles.
- Compatible con otras bibliotecas de componentes, como JBoss RichFaces.
- Uso de *javascript* no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque `<script>`).
- Proyecto *open source*, activo y bastante estable entre versiones.

El autor de esta investigación cree conveniente la utilización de JSF 2.2.5 junto con Primefaces 5.3 para lograr agilizar el proceso de desarrollo del *portlet* además de la utilización de los componentes y filtros que brinda para lograr interfaces enriquecidas para el usuario final.

Eclipse JEE Mars

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 (*integrated development environment*) genérico,

aunque goza de mucha popularidad entre la comunidad de desarrolladores del lenguaje Java usando el *plugin-in* JDT (Java *development tools*) que viene incluido en la distribución estándar del IDE (GENBETA.DEV, 2014).

Principales características

Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una preconfiguración de ventanas y editores, relacionadas entre sí, y que permiten trabajar en un determinado entorno de trabajo de forma óptima (GENBETA.DEV, 2014).

Gestión de proyectos: el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente, documentación, ficheros configuración, árbol de directorios. El IDE proporciona asistentes y ayudas para la creación de proyectos. Por ejemplo, cuando se crea uno, se abre la perspectiva adecuada al tipo de proyecto que se genera, con la colección de vistas, editores y ventanas preconfigurada por defecto (GENBETA.DEV, 2014).

Depurador de código: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente ayuda a mejorar nuestro código. Para ello sólo se debe ejecutar el programa en modo depuración (con un simple botón). De nuevo, se observa una perspectiva específica para la depuración de código, la perspectiva depuración, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.

Extensa colección de *plug-in*: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Cuenta con una colección disponible muy grande. Los hay gratuitos, de pago, bajo distintas licencias, casi para todo tipo se encuentra el *plug-in* adecuado (GENBETA.DEV, 2014).

Liferay IDE 6.2

El Liferay IDE es una extensión de Eclipse que permite crear *plug-in* en un entorno integrado para Liferay. El Liferay IDE de Eclipse permite el desarrollo de *portlets*, *layouts*, temas, *hooks*, y *plug-in* en modalidad EXT. Es importante tener en cuenta que el desarrollador tiene ahora una herramienta de desarrollo especial utilizable en Eclipse, uno de los IDE más populares para el desarrollo de Java (SCAMARIO, 2010).

Apache Tomcat 7.0.42

Tomcat es un contenedor de servlets más utilizados por Liferay Portal, que se utiliza en la referencia oficial de la implementación para Java *Servlet* y (JSP) Java *Server Pages*. Las especificaciones Java *Servlet* y Java *Server Pages* son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java *Community Process*). Apache Tomcat es

desarrollado en un entorno abierto y participativo, bajo la licencia de Apache *Software License* (DÍAZ, 2010).

Tomcat es mantenido y desarrollado por miembros de la *Apache Software Foundation* y voluntarios independientes. Los usuarios disponen de libre acceso al código fuente y forma binaria en los términos establecidos en la *Apache Software License*. Apache Tomcat es un *software* que sirve como servidor web con soporte de *servlets* y JSPs (DÍAZ, 2010). Tomcat es uno de los servidores web más utilizados en el centro (CDAE), por ser un *software* libre, cuenta con una amplia documentación y una gran comunidad de usuarios. Por lo antes expuesto el autor de esta investigación utiliza el mismo para formar parte de la solución a la problemática establecida como servidor web.

Metodología de desarrollo

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del *software*.

La finalidad de una metodología de desarrollo es garantizar la eficacia (cumplir los requisitos iniciales) y la eficiencia (minimizar las pérdidas de tiempo) en el proceso de generación de *software* (BLANCO-CUARESMA, 2008).

Metodología de desarrollo. Metodología Unificada Ágil (AUP).

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de *Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas que incluye Desarrollo Dirigido por Pruebas (*test driven development* - TDD), modelado Ágil, gestión de cambios ágil, y refactorización de base de datos para mejorar la productividad.

El proceso unificado (*Unified Process* o UP) es un marco de desarrollo de *software* iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto de *software*. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (*Rational Unified Process*) de IBM (*International Business Machines*) (LIC. ERVIN FLORES, 2014).

AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura

para los requisitos clave del producto y que determinan los riesgos técnicos (LIC. ERVIN FLORES, 2014).

AUP establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados: (LIC. ERVIN FLORES, 2014)

- Incepción (Concepción): El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- Transición: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Ventajas que brinda la metodología AUP: (LIC. ERVIN FLORES)

- 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 del uso de la metodología AUP: (LIC. ERVIN FLORES)

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

El autor de esta investigación escoge la Metodología Unificada Ágil para guiar el proceso de desarrollo de la solución propuesta a la problemática general, realizándole una modificación de manera que se ajuste a la actividad productiva de la UCI (ciclo de vida).

Modificaciones de (AUP) para la UCI

AUP propone cuatro fases: (Incepción, Elaboración, Construcción, Transición), la UCI mantiene la fase de Incepción como primera para el ciclo de vida sus proyectos. Unificando las tres restantes fases de AUP (Elaboración, Construcción, Transición) en solo una, denominada Ejecución.

1.4 Seguridad informática y Auditoría

La seguridad informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. De todas formas, no existe ninguna técnica que permita asegurar la inviolabilidad de un sistema.

Un sistema informático puede ser protegido desde un punto de vista lógico (con el desarrollo de *software*) o físico (vinculado al mantenimiento eléctrico, por ejemplo) (GALVIS *et al.*, 2009).

Entre los procesos fundamentales en la construcción de un portal intervienen los mecanismos de seguridad.

Mecanismos básicos de seguridad

En general, la seguridad de las redes de datos requiere para conceder acceso a los servicios de la red, tres procesos: (CELY, 2006).

- **Autenticación:** el proceso por el cual el usuario se identifica en forma inequívoca; es decir, sin duda o equivocación de que es quien dice ser.
- **Autorización:** el proceso por el cual la red de datos autoriza al usuario identificado a acceder a determinados recursos de la misma.
- **Registro:** el proceso mediante el cual la red registra todos y cada uno de los accesos a los recursos que realiza el usuario, autorizado o no.

Administración: establece, mantiene y elimina las autorizaciones de los usuarios del sistema, los recursos del sistema y las relaciones usuarios-recursos del sistema.

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. La administración de la seguridad informática dentro de la organización es una tarea en continuo cambio y evolución, ya que las tecnologías utilizadas cambian rápidamente y con ellas los riesgos (MIFSUD, 2012).

Auditoría

Las auditorías llevan a cabo un control acerca del uso correcto de los recursos que brindan las instituciones, la tendencia de los usuarios, detectar sus vulnerabilidades, prevenir errores y fraudes, así como emitir una opinión acerca de la marcha de la empresa (GALVIS *et al.*, 2009).

Auditoría informática

La Auditoría Informática es el proceso de recoger, agrupar y evaluar evidencias para determinar si un sistema informatizado salvaguarda los activos, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos. De este modo la auditoría informática sustenta y confirma la consecución de los objetivos tradicionales de la auditoría:

- Objetivos de protección de activos e integridad de datos.

- Objetivos de gestión que abarcan, no solamente los de protección de activos, sino también los de eficacia y eficiencia (ROLDÁN, 2013).

Auditoría de Sistemas de Información

Es un examen y validación del cumplimiento de los controles y procedimientos utilizados para la confidencialidad, integridad y disponibilidad de los sistemas de información. Determinar si los controles implementados son eficientes y suficientes, identificar las causas de los problemas existentes en los sistemas de información y a su vez las áreas de oportunidad que puedan encontrarse, determinando las acciones preventivas y correctivas necesarias para mantener a los sistemas de información confiables y disponibles (INFORMÁTICA, 2001).

Auditoría y registro

Se define la auditoría como la continua vigilancia de los servicios en producción y para ello se recaba información y se analiza. 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. Se analiza el registro como el mecanismo por el cual cualquier intento de violar las reglas de seguridad establecidas queda almacenado en una base de eventos para luego analizarlo. Pero auditar y registrar no tiene sentido sino van acompañados de un estudio posterior en el que se analice la información recabada. Monitorear la información registrada o auditar se puede realizar mediante medios manuales o automáticos, y con una periodicidad que dependerá de lo crítica que sea la información protegida y del nivel de riesgo (MIFSUD, 2012).

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.

Dentro de las técnicas más utilizadas para mantener (o controlar) la integridad de los datos están: uso de antivirus, encriptación y funciones *'hash'* (MIFSUD, 2012).

Una auditoría realizada a los eventos generados por los usuarios dentro de un portal, le brindaría una serie de información relevante y concisa a los administradores del sistema como:

- Obtención de un criterio confiable a la hora realizar o emprender una modificación al portal.
- Determinar cuál es la media de la aceptación del portal por parte de los usuarios.
- Identificar anomalías o mal uso de los recursos y medios disponibles en el portal, identificando violaciones en las políticas de seguridad.

1.5 Conclusiones

Luego de realizar un estudio detallado acerca de las principales tendencias en el mundo actual referente a tecnologías, metodologías, arquitecturas, lenguajes y herramientas, las cuales servirán como base para el desarrollo de la propuesta final (*portlet* de gestión de trazas y eventos).

- Se realizó un análisis documental relacionado a disciplinas como la seguridad informática, la auditoría, el desarrollo de portales web, *hooks*, *Service Builder* (este utiliza los *frameworks* Spring 3.0.7 e Hibernate 5.0.6) el cual será de gran apoyo para lograr agilizar y optimizar el proceso de captura y almacenamiento en la base de datos de las trazas y eventos generados ya que es una herramienta propia del Liferay Portal.
- La metodología a utilizar (Metodología AUP).
- Como servidor de aplicación web Apache Tomcat 7.0.42.
- Se conformará el *portlet*, utilizando el lenguaje de programación a utilizar java, con un entorno de desarrollo Eclipse Mars 4.5.0 ya que este soporta el *plugin-in* Liferay IDE 6.2 integrado con Maven 3.1.0.
- Conceptos como el de *Framework* JSF 2.2.5 y Primefaces 5.3 también se indicaron para reafirmar el empleo de estos a la hora de crear las interfaces visuales del componente.

Este conjunto de herramientas posibilita la construcción de un producto eficiente, eficaz, robusto y agradable que le permitirá a los desarrolladores fiabilidad en la utilización del Liferay Portal (CE) para desarrollar portales permitiendo almacenar en una base de datos, documento tipo excel o pdf la información de las acciones de los usuarios dentro del sistema.

CAPÍTULO 2 Componente para la gestión de trazas y eventos del Liferay Portal

Con la elaboración de un componente para la gestión de trazas y eventos del Liferay Portal se ayuda de forma fiable a la toma de decisiones de los administradores de los portales web. En el presente capítulo se presenta la propuesta de solución a las insuficiencias por las cuales se generó esta investigación. Se describen y explican las características propias del componente, los diagramas para ayudar su comprensión, así como la arquitectura propuesta para la elaboración del mismo, utilizando como base la metodología AUP.

2.1 Propuesta de solución

Liferay Portal (CE) hoy en día no dispone de un componente modular que posibilite la gestión de las trazas y eventos realizados por los usuarios dentro de los portales web desarrollados a través del mismo. Para establecer una solución a este problema se plantea generar un *portlet* que almacene las trazas de los usuarios dentro del sistema (sin lograr modificar el comportamiento del portal) para lograr mayor fiabilidad en la toma de decisiones de los administradores de los portales.

Con la utilización del componente, el administrador del sitio puede realizar un estudio detallado acerca de la usabilidad del sistema, almacenando las acciones de los usuarios, conformadas por usuario, evento, dirección ip, dominio, descripción del evento, día y hora; así como realizar reportes (salvar) a documentos con formato pdf o excel. También contará con filtros que permiten acotar la búsqueda de cierto tipo de dato, además le permitirá visualizar el comportamiento de los temas analizados a través de un tablero de control con gráficos estadísticos.

Partiendo de este conjunto de restricciones de negocio, con la implementación y puesta en marcha del componente, los administradores de sistemas obtendrán información fiable referente a la utilización de los recursos y herramientas que brinda el portal a los usuarios.

2.2 Ingeniería del Software basada en componentes (ISBC)

Para lograr la correcta elaboración del componente, la Ingeniería del *Software* basada en componentes proporciona métodos, modelos y técnicas para ser utilizadas por los desarrolladores, además expone las principales características que deben de contener los componentes.

Características para que un elemento sea catalogado como componente: (ROJAS y GARCÍA, 2004)

- **Identificable:** Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación.
- **Auto contenido:** Un componente no debe requerir de la utilización de otros para finiquitar la función para la cual fue diseñado.
- **Puede ser remplazado por otro componente:** Se puede remplazar por nuevas versiones u otro componente que lo suplante y mejore.
- **Con acceso solamente a través de su interfaz:** Debe asegurar que estas no cambiarán a lo largo de su implementación.
- **Sus servicios no varían:** Las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí.
- **Bien Documentado:** Un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros, adaptarlo, etc.
- **Es genérico:** Sus servicios deben servir para varias aplicaciones.
- **Reutilizado dinámicamente:** Puede ser cargado en tiempo de ejecución en una aplicación.
- **Independiente de la plataforma:** *Hardware*, *Software* y sistema operativo (S.O.).

Un componente además de contar con todas las características anteriormente descritas, debe de tener una serie de especificaciones para su correcto funcionamiento e integración con la plataforma.

Especificaciones para *portlet*s java JSR 168 y JSR 286

El componente de gestión de trazas y eventos debe cumplir con los estándares para *portlet*s Java *Portlet Specifications* (JSR) 168, el cual define mecanismos para que el *portlet* acceda a los datos transitorios y persistentes. Puede colocar y obtener estos datos en los si siguientes escenarios: (LEON, 2011)

- *Request:* la petición tiene datos incluidos, como parámetros y atributos, puede contener propiedades para permitir que la extensión y los encabezados del cliente sean transportados del portal al *portlet* y recíprocamente.
- *Session:* el *portlet* puede guardar datos en la sesión con alcance global, para dejar que otros componentes de la aplicación web tengan acceso a los datos, o en el alcance del *portlet*, el cual es de acceso restringido al mismo.
- *Context:* el *portlet* puede guardar los datos en el contexto de la aplicación web.

El estándar JSR 286 plantea que los *portlets* puedan compartir datos relacionados con el período de sesiones. El JSR 286 introduce el modelo de eventos tales como:

- Un *portlet* puede declarar eventos que quiere emitir y desea recibir.
- El contenedor de *portlet* actuará como intermediario y distribuirá los eventos en consecuencia.
- Permite la conexión de *portlet* en tiempo de ejecución.

2.3 *Service Builder*

El *Service Builder* es una herramienta de generación de código basado en modelos construidos por Liferay como plataforma de desarrollo. Permite a los desarrolladores definir modelos de objetos personalizados llamados entidades. Genera una capa de servicio a través de la tecnología relacional-objeto de asignación (ORM) que proporciona una separación limpia entre el modelo de objetos y el código de la base de datos subyacente.

El *Service Builder* toma un archivo XML como entrada y genera el modelo necesario, la persistencia, y las capas de servicio para su aplicación. Además, genera la mayor parte del código común necesario para implementar, crear, leer, actualizar, eliminar y buscar operaciones en la base de datos, lo que le permite centrarse en los aspectos de diseño de alto nivel de servicio.

Principales beneficios del uso del *Service Builder*: (LIFERAY, DEVELOPER NETWORK, 2014b)

- Integración con Liferay.
- Generación automática de los modelos, las capas de persistencia y de servicios.
- Generación automática de los servicios locales y remotos.
- Generación automática de las configuraciones de Spring e Hibernate.
- Soporte para generar métodos de búsqueda de entidades.
- Soporte integrado de almacenamiento en caché de la entidad.
- Soporte a consultas SQL personalizadas y dinámicas guardadas en tiempo de ejecución.
- Ahorro de tiempo.

Configuración del *Service Builder*:

- Es necesario crear un archivo `service.xml` y ejecutar el servicio generador.
- El *Service Builder* genera un archivo `jar` para el proyecto. El archivo incluye una capa de modelo, una capa de persistencia, una capa de servicio y la infraestructura relacionada.
 - La capa del modelo es responsable de definir los objetos para representar las entidades de su proyecto.

- La capa de persistencia es responsable de salvar entidades y obtención de las entidades de la base de datos.
- La capa de servicio es la encargada de exponer CRUD y métodos relacionados para sus entidades como una API.

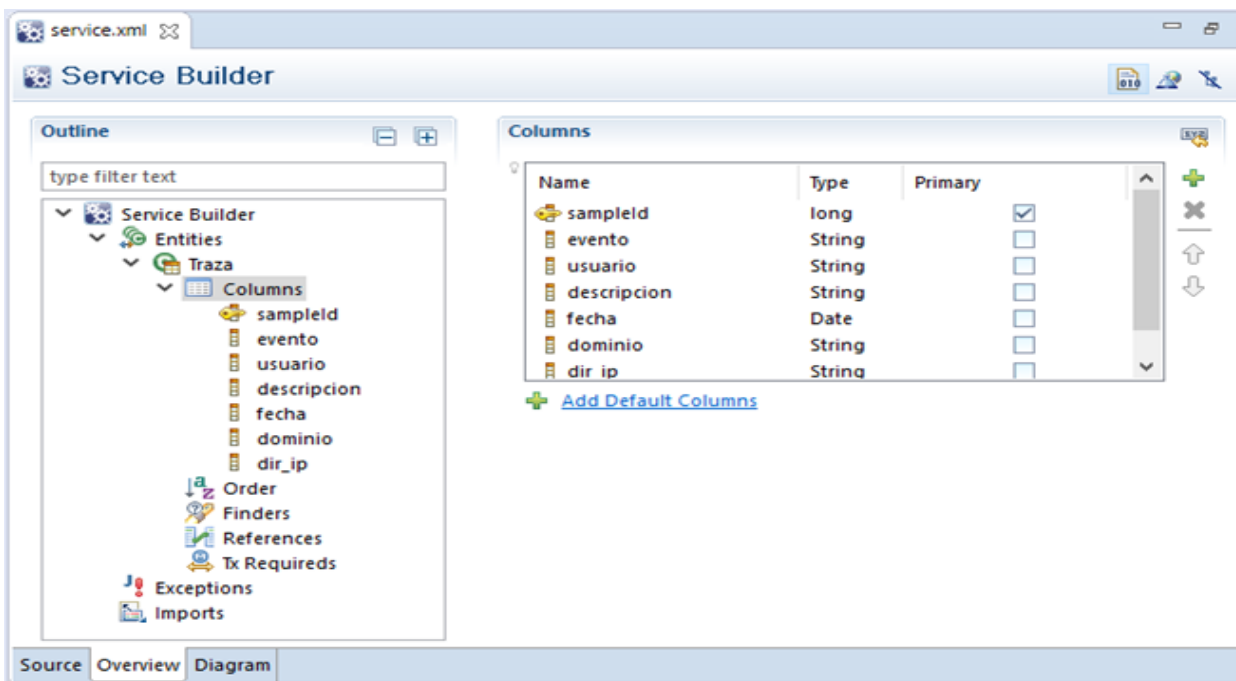


Ilustración 4: Archivo service.xml. Elaboración propia.

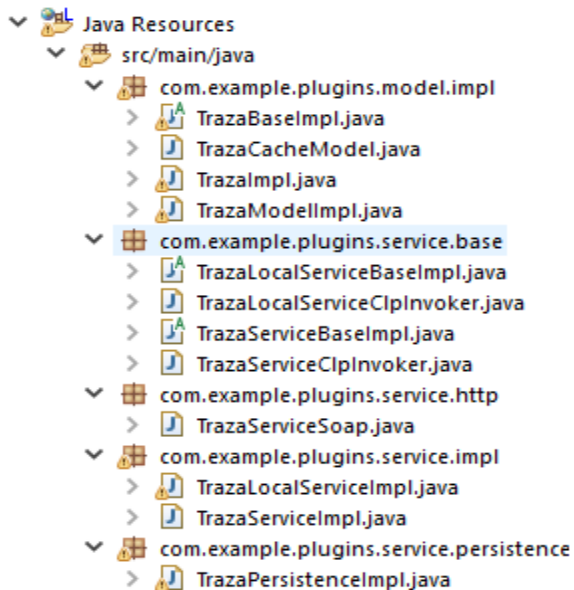


Ilustración 5: Estructuras de los paquetes generados por el service builder. Elaboración propia.

Cada entidad generada por el *Service Builder* contiene una clase de implementación del modelo y una clase local de implementación del servicio configurado en el archivo *service.xml*. Las personalizaciones y la lógica del negocio se pueden implementar en estas tres clases; estas son las únicas clases generadas por el *Service Builders* que están destinadas a ser personalizadas.

2.4 Estructura del componente

El componente es generado a través de un proyecto multi-módulo para evitar implementar todas las clases java y ficheros de configuración directamente en el proyecto web. Para la conformación de estos tipos de proyectos maven en el *pom* del proyecto padre hace referencia a los proyectos hijos.

El componente está estructurado en tres módulos jar (*Java Archive*), SB-P-PF (proyecto padre), SB-P-PF-*portlet* (proyecto hijo) y por último SB-P-PF-*portlet-service*, que una vez desplegado el proyecto maven se encapsulan en el war (*Web Application Archive*).

El war generado incluye todas las dependencias y herramientas para que el componente pueda desplegarse en cualquier servidor Liferay Portal superior a la versión 6.2. Con solo cargar el archivo en el servidor se autogenera toda la infraestructura para el correcto funcionamiento del componente, la capa de lógica del negocio, el acceso a datos y la propia base de datos para almacenar la información, logrando la portabilidad del componente.

Importancia de la utilización de maven: (GARCÍA, 2014)

- Maven se basa en patrones y estándares. Esto permite a los desarrolladores moverse entre proyectos y no necesitan aprender como compilar o empaquetar. Mejora el mantenimiento y la reusabilidad.
- El *portlet* se compone (nombre, versión, bibliotecas de las que depende, etc), y maven se encargará de hacer todas las tareas por nosotros (incluir, descargar dependencias, entre otras).
- Maven hace la gestión de librerías, ocupándose de dependencias transitivas. Es decir, si A depende de B y B depende de C, es que A depende de C. Esto quiere decir que cuando empaqueta A, maven se encargará de añadir tanto B como C en el paquete.

El autor de esta investigación define la utilización de maven por las prestaciones que brinda para el desarrollo del componente principalmente por la portabilidad que le otorga al *portlet*.

2.5 Requisitos del sistema

Los requisitos del sistema se obtuvieron a través de un levantamiento de las principales funcionalidades que deberá cumplir el componente. El autor de esta investigación realizó entrevistas a jefes de proyectos y desarrolladores de portales pertenecientes al CDAE con

experiencia en el uso del Liferay. Consultándoles las deficiencias que les ocasiona la ausencia de un componente capaz de gestionar las trazas y eventos del Liferay Portal a la hora de la toma de decisiones. Se realizaron observaciones de la información que monitorean otras aplicaciones que se pueden utilizarse para el desarrollo de portales como Joomla, Drupal.

Los requisitos funcionales y no funcionales son:

2.5.1 Requisitos funcionales (RF)

No	Nombre	Descripción	Complejidad
[RF1.]	Registrar eventos de los usuarios.	El sistema debe ser capaz una vez que el usuario realice una acción, generar un evento y registrarlo en la base de datos.	Alta
[RF2.]	Obtener listado de los eventos de los usuarios.	El sistema debe permitir obtener los eventos existentes en la base de datos generados por las acciones de los usuarios.	Alta
[RF3.]	Filtrado por campos.	El sistema debe permitir filtrar (acotar) la búsqueda por un valor determinado de un evento.	Media
[RF4.]	Reporte (exportar) a formato pdf o excel.	El sistema debe permitir generar documentos tipo pdf o excel para exportar toda la información referente a las trazas de los usuarios.	Media
[RF5.]	Tablero de control para visualizar el comportamiento de los temas analizados.	El sistema debe permitir visualizar a través de gráficos estadísticos el comportamiento de las conexiones de los usuarios.	Baja
[RF6.]	Visualizado solo por los administradores del portal.	El componente solo se debe visualizar desde el panel de control al cual solo tienen acceso los administradores del portal.	Baja

Tabla 1: Tabla de Requisitos funcionales.

Los temas (indicadores) visualizados a través de gráficos son:

- Conexiones por dominios: muestra los cuatro dominios que más visitan el portal.
- Páginas más visitadas: muestra las páginas que más aceptación tienen por los usuarios.
- Diarias: muestra la cantidad de conexiones *Login* y *Logout* cada tres horas por tiempo máximo de un día, el actual.
- Últimos 7 días: muestra la cantidad de conexiones *Login* y *Logout* que han realizados los usuarios en el portal en los últimos siete días contando como último el día actual.

2.5.2 Requisitos no funcionales (RNF)

No	Nombre	Descripción
[RNF1.]	Soporte	Utilizar como servidor de base de datos PostgreSQL Server 9.2.x, Oracle o MySQL.
[RNF2.]	Soporte	El sistema debe ejecutarse en navegadores Mozilla Firefox 30.0, Internet Explorer 7, Chrome 30.0, Safari u Opera 10 o en otro superior a los mencionados.
[RNF3.]	Soporte	Utilizar como servidor web Apache Tomcat 7.0.x, Jboss o GlassFish.

Tabla 2: Tabla de Requisitos no funcionales.

2.6 Historias de Usuario

La metodología escogida para guiar el desarrollo del componente (AUP), emplea las historias de usuario para la descripción de una funcionalidad que debe incorporar un *software*, y cuya implementación aporta valor al cliente.

La estructura de una historia de usuario está formada por: (BOK, 2014)

- Nombre breve y descriptivo.
- Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar.
- Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Y adicionalmente por la información que resulte necesaria por el modelo de implementación: Prioridad, Riesgo y Tamaño.

Número: Historia de Usuario 1	Nombre del requisito: Registrar eventos de los usuarios.
Programador: Brian Valdés Martínez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 2 días

Riesgo en Desarrollo: <ul style="list-style-type: none"> • Problemas eléctricos. • Problemas técnicos de los servicios de redes. 	Tiempo Real: 18 horas
Descripción: El sistema debe ser capaz una vez que el usuario realice una acción, generar un evento y registrarlo en la base de datos.	
Observaciones:	
Prototipo de interfaz:	

Tabla 3: HU Registrar eventos de los usuarios.

Número: Historia de Usuario 2	Nombre del requisito: Obtener listado de los eventos de los usuarios.
Programador: Brian Valdés Martínez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Problemas eléctricos. • Problemas técnicos de los servicios de redes. 	Tiempo Real: 18 horas
Descripción: El sistema debe permitir obtener los eventos existentes en la base de datos generados por las acciones de los usuarios.	
Observaciones:	
Prototipo de interfaz:	

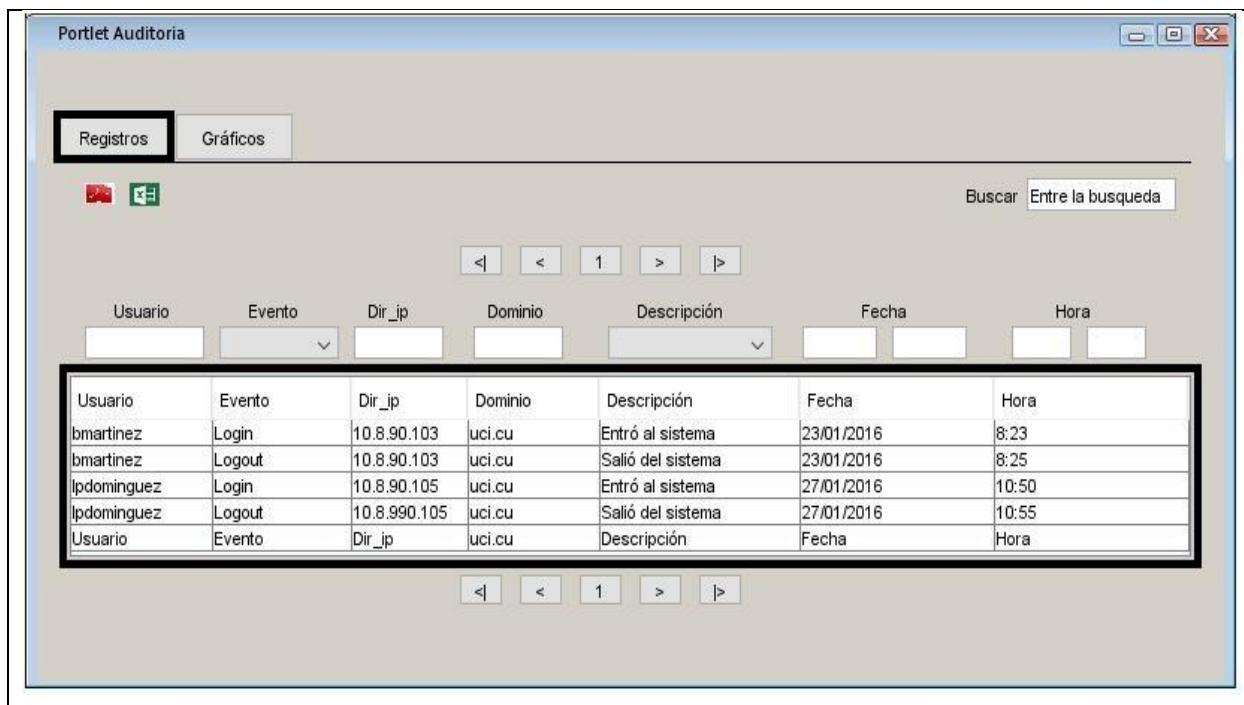


Tabla 4: HU Obtener listado de los eventos de los usuarios.

Número: Historia de Usuario 3	Nombre del requisito: Filtrado por campos.
Programador: Brian Valdés Martínez	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 5 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Problemas eléctricos. • Problemas técnicos de los servicios de redes. 	Tiempo Real: 40 horas
Descripción: El sistema debe permitir filtrar (acotar) la búsqueda por un valor determinado de un evento.	
Observaciones:	
Prototipo de interfaz:	

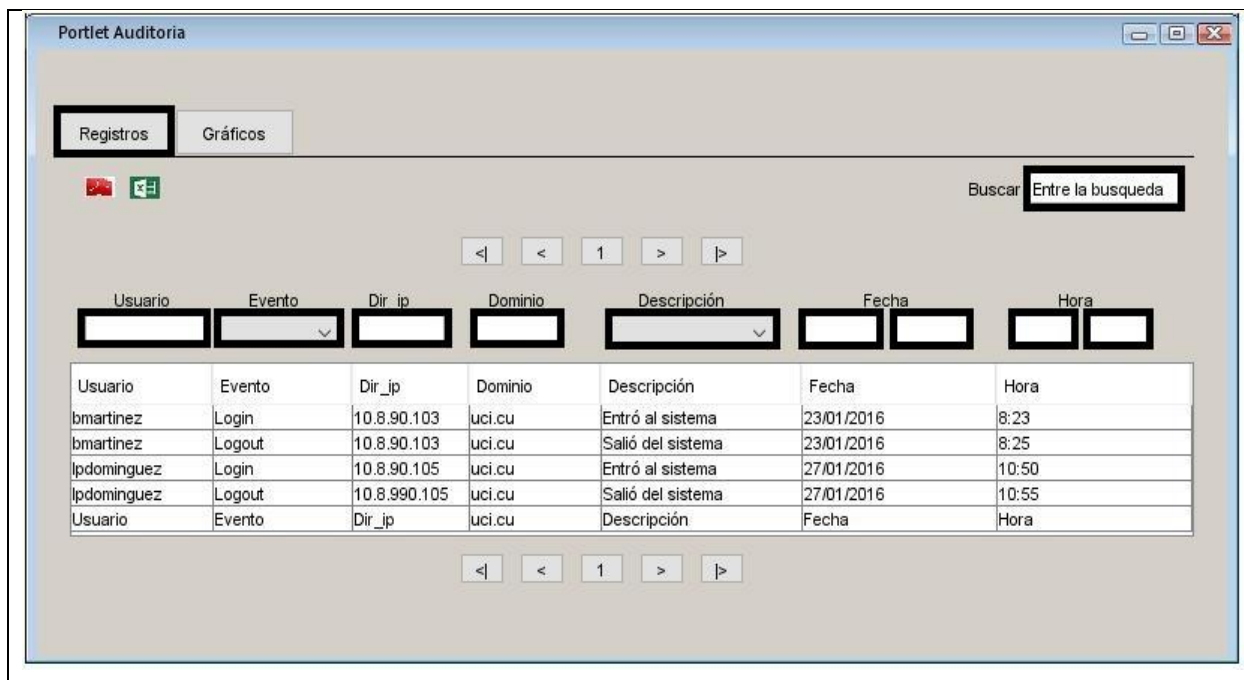


Tabla 5: HU Filtrado por campos.

Número: Historia de Usuario 4	Nombre del requisito: Reporte (exportar) a formato pdf o excel.
Programador: Brian Valdés Martínez	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 1 día
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Problemas eléctricos. • Problemas técnicos de los servicios de redes. 	Tiempo Real: 6 horas
Descripción: El sistema debe permitir generar documentos tipo pdf o excel para exportar toda la información referente a las trazas de los usuarios.	
Observaciones:	
Prototipo de interfaz:	

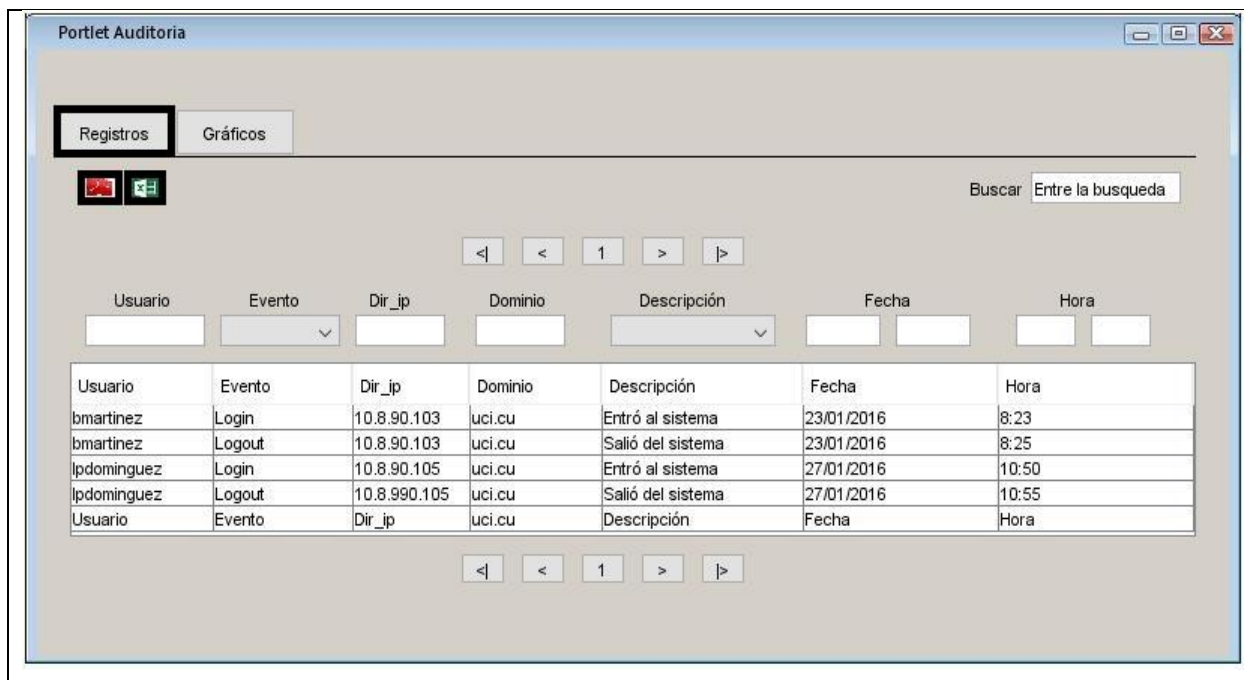


Tabla 6: HU Reporte (exportar) a formato pdf o Excel.

Número: Historia de Usuario 5	Nombre del requisito: Tablero de control para visualizar el comportamiento de los temas analizados.
Programador: Brian Valdés Martínez	Iteración Asignada: 1
Prioridad: Baja	Tiempo Estimado: 2 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Problemas eléctricos. • Problemas técnicos de los servicios de redes. 	Tiempo Real: 14 horas
Descripción: El sistema debe permitir visualizar a través de gráficos estadísticos el comportamiento de las conexiones de los usuarios.	
Observaciones:	
Prototipo de interfaz:	

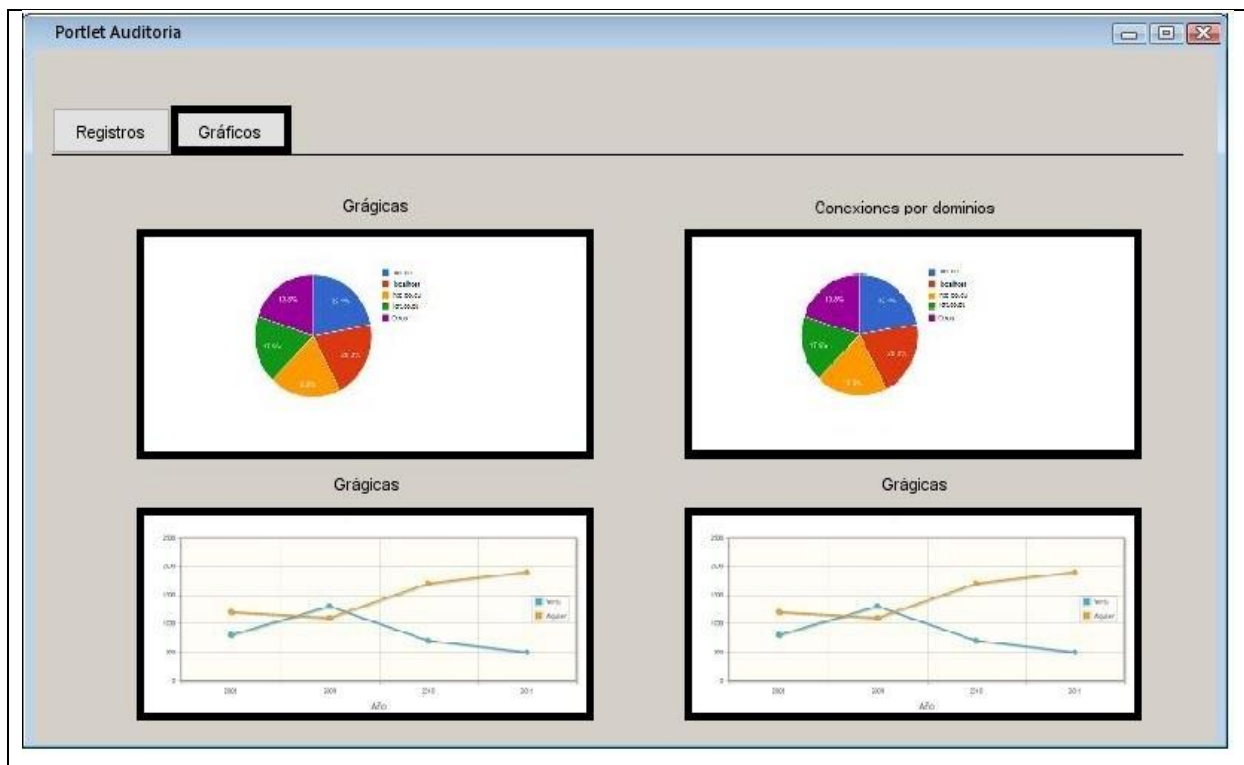


Tabla 7: HU Visualizar el comportamiento de las conexiones.

Número: Historia de Usuario 6	Nombre del requisito: Visualizado solo por los administradores del portal.
Programador: Brian Valdés Martínez	Iteración Asignada: 1
Prioridad: Baja	Tiempo Estimado: 1 días
Riesgo en Desarrollo: <ul style="list-style-type: none"> • Problemas eléctricos. • Problemas técnicos de los servicios de redes. 	Tiempo Real: 6 horas
Descripción: El componente solo se debe visualizar desde el panel de control al cual solo tienen acceso los administradores del portal.	
Observaciones:	
Prototipo de interfaz:	

Tabla 8: HU Tablero de control para visualizar el comportamiento de los temas analizados.

2.7 Modelo de dominio

Un modelo de dominio es una representación visual de las clases conceptuales o de objetos reales del dominio de interés, es un diagrama de clases sin la identificación de operaciones. Permitiendo una mayor comprensión para los desarrolladores, clientes e involucrados con el componente en función.

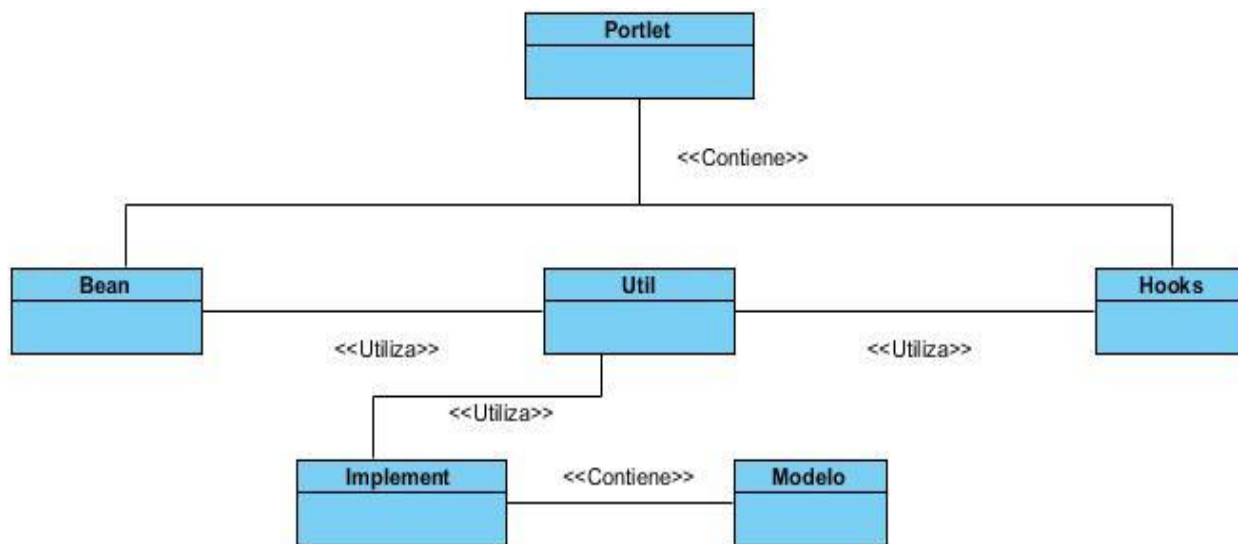


Ilustración 6: Modelo de dominio. Elaboración propia.

Descripción del modelo de dominio.

Nombre	Descripción
Portlet	Contiene la vista del <i>portlet</i> .
Hooks	Clase en la cual se generan y adicionan las trazas.
Bean	Es la clase que gestiona todo el flujo de información.
Útil	Clase que se instancia para acceder a los métodos implementados para lograr el acceso a los datos como almacenar y adicionar.
Implement	Clase en la cual se implementan los métodos insertar y listar trazas.
Modelo	Modelo traza, la cual tiene todos los atributos y constructores de un evento.

Tabla 9: Descripción del modelo conceptual.

2.8 Estilo arquitectónico y patrones utilizados

Para lograr una mayor comprensión acerca del diseño y contenido del componente se muestra una descripción detallada del sistema para guiar la implementación.

Arquitectura de la aplicación

El autor de esta investigación apoyándose en los requisitos funcionales e historias de usuarios define para la confección del componente de gestión de trazas una arquitectura por capas:

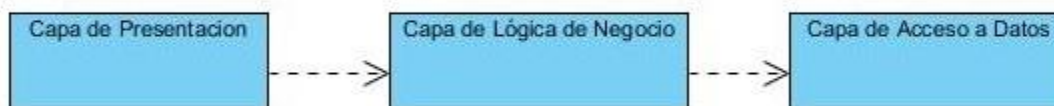


Ilustración 7: Arquitectura en 3 Capas. Elaboración propia.

- **Capa de presentación:** Contiene las interfaces que conforman la aplicación, las cuales son presentadas al usuario como programa final, deben ser amigables e intuitivas. Para la generación de la interfaz del proyecto se empleó el *framework* JSF y Primefaces, utilizando el patrón Modelo Vista Controlador. Contando con la implementación de un *bean* para resolver todo lo referente al flujo de información que se le mostrará al administrador del sistema. La clase del proyecto que genera la vista se denomina `view.xhtml`.
- **Capa de lógica del negocio:** Es donde se establecen las reglas que deben cumplir, las acciones tanto del usuario como del sistema. Recibe las peticiones por parte del usuario y genera las respuestas a las mismas, empleando Spring.
- **Capa de acceso a datos:** Responde a las peticiones de la capa lógica mediante el uso del patrón (DAO) Objetos de Acceso a Datos, utilizando como *framework* Hibernate.

Las capas de **Lógica del negocio** y **Acceso a datos** están embebidas dentro del paquete generado por la herramienta **Service Builder** del **Liferay Portal (CE)** (`TrazaLocalServiceImpl.java`).

Patrón Modelo Vista Controlador (MVC):

El patrón arquitectónico modelo vista controlador es una propuesta de diseño de *software* utilizada para desarrollar sistemas donde interviene la utilización de interfaces de usuario. Emerge de la necesidad de crear *softwares* robustos con un ciclo de vida más adecuado, donde se fortalezca la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Modelo:

Es la capa donde se trabaja con los datos, contiene los mecanismos para acceder a la información y también actualizar su estado. La información estará almacenada en una base de datos, por lo que en los modelos se encuentran todas las funciones con las cuales acceder a las tablas, actualizar, seleccionar y adicionar los datos.

En el componente de gestión de trazas, la generación del modelo y todo lo referente a la lógica del negocio y el acceso a datos es automáticamente generado mediante el *Service Builder*, el cual utiliza el (service.xml) donde están todos los datos necesarios, nombres de tablas, atributos, entre otros.

Vistas:

Las vistas, contienen el código de la aplicación, permitiendo la visualización de las interfaces de usuario, o sea, el código que permite mostrar los estados de la aplicación. En ellas solo se encuentran los códigos HTML que muestran la salida.

La clase que representa la vista de la aplicación es (view.xhtml), la cual con la utilización de Primefaces hace del componente un producto más amigable, agradable e interesante para el usuario final.

Controladores:

En ellos está contenido el código necesario para responder a las acciones que el usuario le solicita a la aplicación, como visualizar los gráficos, realizar un reporte, una búsqueda de información, entre otros.

Para generar el correcto procesamiento de la información se han establecido varias clases controladoras:

TrazaBean: clase en la cual se interconectan el modelo y la vista, haciendo de intermediario entre las peticiones del usuario y las acciones predefinidas del sistema.

ChartBean y PieBean: clases en las cuales se generan las gráficas de líneas, barras y pasteles respectivamente, los cuales muestran información relevante para el administrador del sistema acerca del comportamiento de los usuarios.

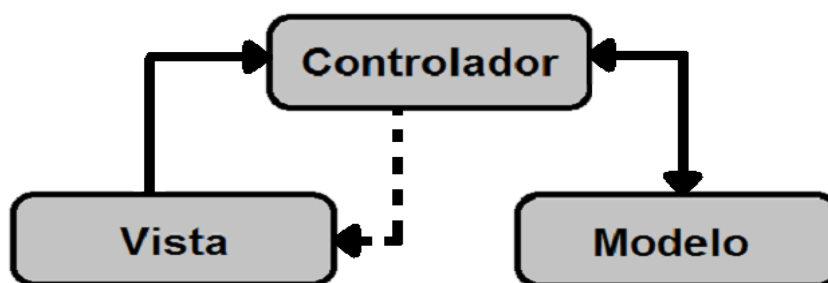


Ilustración 8: Patrón Modelo Vista Controlador. Elaboración propia.

Patrones de diseño

- **DAO (*Data Access Object*):** la funcionalidad principal del mismo es contar con diversas fuentes de datos (entre ellas se encuentran archivos y bases de datos). Encapsula la forma de acceder a los datos. Surge por la necesidad de gestionar una variedad de información, no solo trata el problema de empaquetar el origen de los datos, también oculta la forma de acceder a ellos. Utilizado a través del *Service Builder*.
- **Composite view:** Normalmente una funcionalidad está formada por un grupo de vistas, siendo este, el patrón que integra varias subvistas en una sola página. Es necesario estandarizar el proceso de integración y composición. Habitualmente diferentes tipos de individuos con disimiles habilidades ayudan al desarrollo y mantenimiento de las páginas. En el Liferay Portal cada componente de la plantilla se puede incluir de forma dinámica en el conjunto, y el diseño de la página se puede manejar independientemente del contenido.

2.9 Conclusiones parciales

En el capítulo se presentó el análisis, comprensión y las características de la solución propuesta para resolver la problemática por la cual se desarrolla esta investigación:

- Se identificaron los requisitos funcionales y no funcionales.
- Se describieron los patrones de diseño empleados, los cuales brindan ventajas con su utilización.
- Se definió la arquitectura por la cual va a estar compuesto el componente para lograr un mayor entendimiento acerca del funcionamiento del mismo.

CAPÍTULO 3 Implementación y Pruebas

En presente capítulo se lleva a cabo el proceso de implementación y pruebas al componente de gestión de trazas y eventos del Liferay Portal, demostrando el cumplimiento de los requisitos funcionales para resolver la problemática de la investigación.

3.1 Estándar de código

Un estándar de código facilita el trabajo en un proyecto, para que cualquier persona involucrada en el mismo tenga acceso y comprenda el código. Define la escritura y organización del código fuente de un programa. Facilita al programador la modificación de su propio código fuente, aunque no esté trabajando en el equipo. Establece la forma en que deben ser declaradas las variables, las clases, los comentarios. Especifica qué datos deben incluirse acerca del programador y de los cambios realizados al código fuente (EMANUEL, 2013).

Variables:

En la declaración de variables se utilizó la variante *lowerCamelCase*. Las cuales empiezan con minúsculas y si están compuestas por más de una palabra, las siguientes comenzarán con mayúscula.

Métodos:

Los nombres de métodos deben iniciar con un verbo.

Ejemplo:

```
public Traza addTraza(...) {...}
```

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

Ejemplo:

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

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

Ejemplo:

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

Constantes:

Las constantes o campos finales son escritos en letras minúsculas.

Ejemplo:

```
range = new DateRange();
```

Clases:

Se hace uso de la variante *UpperCamelCase* para los identificadores. Las palabras que distinguen a los mismos empezarán con mayúscula.

Ejemplo:

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

Comentarios de las clases:

Con el uso de los comentarios en bloque al inicio de cada archivo y antes de cada método se logra proporcionar una breve descripción de los mismos.

Ejemplo:

```
/**
 *
 * Listar todas las trazas insertadas
 *
 * @return*
 */
```

Vistas del Componente de Gestión de trazas y eventos del Liferay Portal

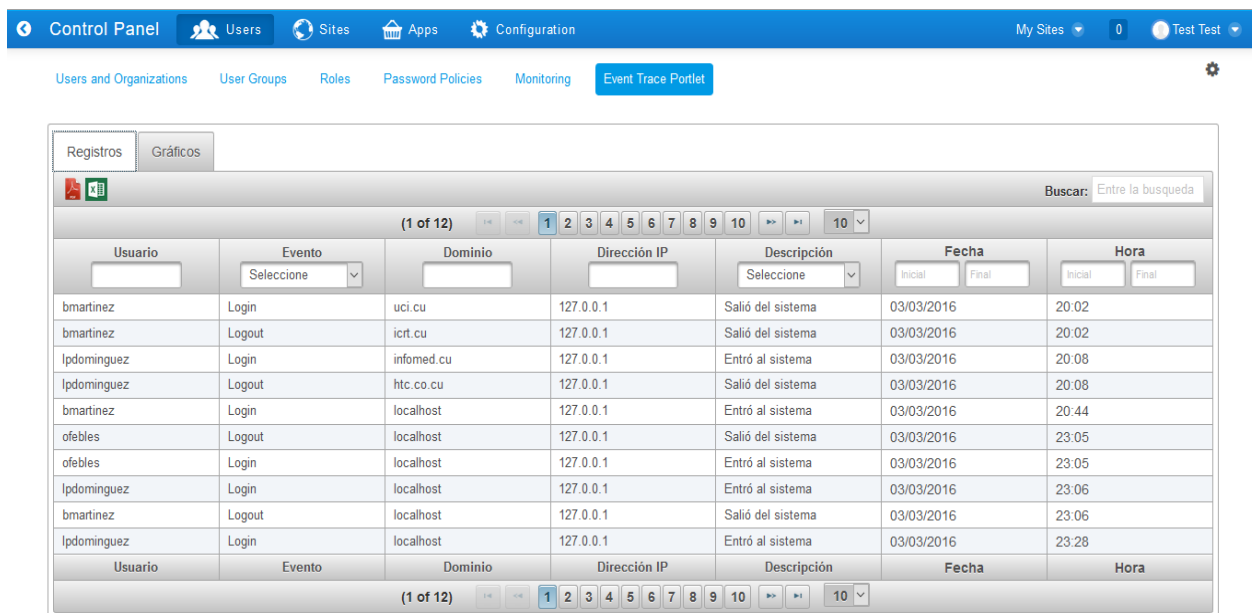


Ilustración 9: Interfaz principal del componente. Elaboración propia.

3.2 Pruebas

Procesos a los que se somete el sistema, los cuales evalúan la calidad del producto, en cuanto a posibles fallos de usabilidad, calidad o implementación en el código. Las pruebas de *software* que se realizan para evaluar el nivel de factibilidad y calidad son las de caja negra a través de técnicas experimentales.

Pruebas de caja negra

- Pruebas funcionales dedicadas a la interfaz de la aplicación.
- Estas pruebas se enfocan principalmente hacia un módulo, o sección específica del *software*. Se limitan a estudiar con datos de entrada cuáles son las salidas.

3.2.1 Estrategia de pruebas

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Registrar eventos de los usuarios.	EC 1.1: Se obtuvo correctamente el registro de los eventos.	El sistema debe ser capaz de una vez el usuario realice una acción, generar un evento y registrarlo en la base de datos.	
SC 2: Obtener listado de los eventos de los usuarios.	EC 2.1: Se obtuvo correctamente el listado de los eventos de los usuarios.	El sistema debe permitir obtener los eventos existentes en la base de datos generados por las acciones de los usuarios.	1- En el portal Liferay, al arrancar el <i>Portlet</i> de gestión de trazas y eventos se debe de listar correctamente los eventos almacenados en la base de datos.

<p>SC 3: Filtrado por campos.</p>	<p>EC 3.1: Acción realizada satisfactoriamente.</p>	<p>El sistema debe permitir filtrar (acotar) la búsqueda por un valor determinado de un evento.</p>	<p>1- En el portal Liferay, en el <i>Portlet</i> de gestión de trazas y eventos, dirigirse campo de entrada del (filtro) correspondiente.</p> <p>2-Dependiendo del criterio de entrada del filtro se introducirá manualmente o por selección.</p> <p>3-Luego de introducir los datos se procederá al acotamiento de la lista, mostrando los datos iguales o que contengan al criterio insertado.</p>
<p>SC 4: Reporte (exportar) a formato pdf o excel.</p>	<p>EC 4.1: Acción obtenida satisfactoriamente.</p>	<p>El sistema debe permitir generar documentos tipo pdf o excel para exportar toda la información referente a las trazas de los usuarios.</p>	<p>1- En el portal Liferay, en el <i>Portlet</i> de gestión de trazas y eventos, dirigirse al icono de pdf o excel.</p> <p>2- Al seleccionar la acción que se desee, aparecerá una ventana emergente el cual brindará las opciones de guardar o ver el documento generado de información almacenada.</p>

<p>SC 5: Tablero de control para visualizar el comportamiento de los temas analizados.</p>	<p>EC 5.1: Acción realizada satisfactoriamente.</p>	<p>El sistema debe permitir visualizar a través de gráficos estadísticos el comportamiento de las conexiones de los usuarios.</p>	<p>1- En el portal Liferay, en el <i>Portlet</i> de gestión de trazas y eventos, dirigirse a la pestaña Gráficos. 2-Una vez seleccionada se mostrarán los gráficos estadísticos.</p>
<p>SC 6: Visualizado solo por los administradores del portal.</p>	<p>EC 6.1: Acción realizada satisfactoriamente.</p>	<p>El componente solo se debe visualizar desde el panel de control al cual solo tienen acceso los administradores del portal.</p>	<p>1- En el portal Liferay, una vez identificado un usuario como administrador del portal tendrá acceso al panel de control. 2-Una vez en el panel de control seleccionar <i>Event Trace Portlet</i> y se muestra la interfaz del componente con todas sus funcionalidades.</p>

Tabla 10: Estrategia de pruebas.

3.2.2 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	Respuesta del Sistema	Resultado de la Prueba
EC 3.1	Filtrado por campos (manual).	V Valor.	Mostrar una lista acotada por el criterio previamente entrado ya sea con el valor igual o parcial; o vacía si no existe el resultado buscado.	Se agrega una acción al sistema.
		NA Nulo.	El sistema mostrará la lista inicial la cual contiene a todos los valores almacenados en la base de datos.	
EC 3.2	Filtrado por campos (selección).	V Selección del criterio.	Mostrar una lista acotada por el criterio previamente seleccionado la cual contendrá exactamente el valor igual al resultado buscado.	Se agrega una acción al sistema.

		NA Nulo.	El sistema debe mostrar la lista inicial la cual contiene a todos los valores almacenados en la base de datos.	
--	--	-------------	--	--

Tabla 11: Caso de prueba EC 3: Filtrado por campos.

3.2.3 Lista de chequeo

Característica de soporte comprobar.	Ponderación por importancia	de 1(poco), 5 (fundamental)	Nivel de cumplimiento (0-100)	de Justificación
1-¿Se utilizó como servidor de base de datos PostgreSQL 9.2 para el desarrollo del sistema?	3		100	Se utilizó el gestor de base de datos PostgreSQL 9.2, por ser uno de los más usados en el CDAE. Posee una abundante documentación. El código fuente se encuentra disponible para todos sin costo alguno. Debido a la liberación de la licencia.
¿El <i>Portlet</i> Administrativo es adaptable a los navegadores Moxilla Firefox 30, Internet Explorer 7, Chrome 30.0, Safari u Opera 10?	5		100	El sistema se desplego satisfactoriamente en diferentes entornos de navegación: Moxilla Firefox, Internet Explorer, Chrome, Safari y Opera.

<p>¿Para desarrollar la aplicación se utilizó como servidor web el Apache Tomcat 7.0?</p>	<p>5</p>	<p>100</p>	<p>Se utilizó el servidor web Apache Tomcat por poseer ayuda y abundante documentación.</p> <p>Es libre y uno de los más usados en el CDAE.</p> <p>Se instaló la Máquina Virtual de Java para trabajar con esta herramienta</p>
---	----------	------------	---

Tabla 12: Lista de chequeo.

Resultados

Fueron necesarias dos iteraciones de las pruebas de caja negra para corroborar el funcionamiento correcto del componente de gestión de trazas y la satisfacción de los requisitos funcionales que debe de cumplir la aplicación. En la primera iteración fueron identificadas tres no conformidades referentes al filtrado por fecha el cual no funcionaba correctamente, filtro seleccionar eventos no filtraba una vez seleccionado un ítem y el buscador global no tenía función asignada, las cuales fueron corregidas por el realizador de la aplicación. En la segunda iteración se encontraron 2 no conformidades el filtro seleccionar eventos no debe filtrar por el primer *ítem* y en el filtro fecha se debe de validar que la fecha inicial sea menor a la final, las cuales fueron corregidas satisfactoriamente.

3.3 Pruebas de aceptación

Las pruebas de aceptación son realizadas para que el cliente certifique que el sistema es válido para él. Son pruebas funcionales realizadas al sistema completo, buscando comprobar que se satisfacen todos los requisitos funcionales. Estas pruebas son elaboradas sobre la base de las historias de usuarios, donde una historia puede tener más de una prueba de aceptación para lograr el correcto funcionamiento del producto.

Los datos a medir en las pruebas de aceptación son:

- **Iteración:** Número de la iteración realizada.
- **Nombre del rasgo a probar:** Nombre del rasgo al que se le realiza la prueba.
- **No. del rasgo a probar:** Número del rasgo al que se le realiza la prueba.
- **Descripción del rasgo a probar:** Descripción breve del rasgo que se prueba.

- **Precondiciones:** Condiciones necesarias para poder realizar el caso de prueba y obtener los resultados esperados.
- **Pasos:** Pasos lógicos a seguir durante el desarrollo de la prueba para la obtención del resultado esperado.
- **Resultados esperados:** Descripción breve de los resultados esperados luego de realizar la prueba.
- **Evaluación:** Clasificación de la prueba dependiendo de la comparación del resultado obtenido con el resultado esperado.

Iteración No.: 1	
Nombre del requisito a probar: Registrar eventos de los usuarios.	Número del requisito a probar: [RF1.]
Descripción del requisito a probar: El sistema debe ser capaz una vez el usuario realice una acción, generar un evento y registrarlo en la base de datos.	
Precondiciones: El usuario debe de realizar una acción determinada, para generar un evento y almacenarlo en la base de datos.	
Pasos: 1- El usuario realiza la acción. 2- Se genera el evento correspondiente e inserta en la base de datos.	
Resultados esperados: Se agrega satisfactoriamente un evento a la base de datos.	
Evaluación: Prueba satisfactoria.	
Observaciones: Cumple con la precondición establecida de almacenar eventos.	

Tabla 13: Pruebas de aceptación [RF1.].

Iteración No.: 1	
Nombre del requisito a probar: Obtener listado de los eventos de los usuarios.	Número del requisito a probar: [RF2.]
Descripción del requisito a probar: El sistema debe permitir obtener los eventos existentes en la base de datos generados por las acciones de los usuarios.	
Precondiciones: Al cargar el componente de gestión de trazas y eventos, debe de cargar correctamente la información de la base de datos en la tabla.	
Pasos: 1- El administrador entra al panel de control y busca el <i>portlet</i> . 2- Al cargar el <i>portlet</i> se lista la información de la base de datos.	

Resultados esperados: Se muestra satisfactoriamente la lista en la interfaz principal del <i>portlet</i> .
Evaluación: Prueba satisfactoria.
Observaciones: Cumple con la precondition establecida de cargar la información en un tabla.

Tabla 14: Pruebas de aceptación [RF2.].

Iteración No.: 1	
Nombre del requisito a probar: Filtrado por campos.	Número del requisito a probar: [RF3.]
Descripción del requisito a probar: El sistema debe permitir filtrar (acotar) la búsqueda por un valor determinado de un evento.	
Precondiciones: El sistema debe permitir filtrar por cada valor de los eventos listados en la tabla.	
Pasos:	
<ol style="list-style-type: none"> 1- El administrador dependiendo del valor por el cual quiera filtrar la información deberá insertar el criterio de búsqueda, manual o por selección. 2- Una vez insertado el criterio de búsqueda, el sistema procede a mostrar la información igual o parcial, en caso de que no exista se mostrará la lista vacía. 	
Resultados esperados: Se filtra satisfactoriamente la lista en la interfaz principal del <i>portlet</i> .	
Evaluación: Prueba satisfactoria.	
Observaciones: Cumple con la precondition establecida de filtrar por cada valor de los eventos listados en la tabla.	

Tabla 15: Pruebas de aceptación [RF3.].

Iteración No.: 1	
Nombre del requisito a probar: Reporte (exportar) a pdf o excel.	Número del requisito a probar: [RF4.]
Descripción del requisito a probar: El sistema debe permitir generar documentos tipo pdf o excel para exportar toda la información referente a las trazas de los usuarios.	
Precondiciones: El sistema debe permitir exportar la información listada en la interfaz principal del <i>portlet</i> .	
Pasos:	
<ol style="list-style-type: none"> 1- El administrador al seleccionar el icono (pdf o excel). 	

2- Aparecerá una ventana emergente el cual brindará las opciones de guardar o ver el documento generado de información almacenada.
Resultados esperados: Un documento que contenga la información listada en la interfaz principal del componente.
Evaluación: Prueba no satisfactoria.
Observaciones: El nombre del documento debe de contener la fecha del día actual y el documento tipo pdf debe de exportar correctamente los caracteres especiales.

Tabla 16: Pruebas de aceptación [RF4.].

Iteración No.: 2	
Nombre del requisito a probar: Reporte (exportar) a pdf o excel.	Número del requisito a probar: [RF4.]
Descripción del requisito a probar: El sistema debe permitir generar documentos tipo pdf o excel para exportar toda la información referente a las trazas de los usuarios.	
Precondiciones: El sistema debe permitir exportar la información listada en la interfaz principal del <i>portlet</i> .	
Pasos: <ol style="list-style-type: none"> 1- El administrador al seleccionar el icono (pdf o excel). 2- Aparecerá una ventana emergente el cual brindará las opciones de guardar o ver el documento generado de información almacenada. 	
Resultados esperados: Un documento que contenga la información listada en la interfaz principal del componente.	
Evaluación: Prueba satisfactoria.	
Observaciones: Cumple con la precondición establecida de exportar la información listada en la interfaz principal del <i>portlet</i> .	

Tabla 17: Pruebas de aceptación [RF4.] 2da iteración.

Iteración No.: 1	
Nombre del requisito a probar: Tablero de control para visualizar el comportamiento de los temas analizados.	Número del requisito a probar: [RF5.]
Descripción del requisito a probar: El sistema debe permitir visualizar a través de gráficos estadísticos el comportamiento de las conexiones de los usuarios.	
Precondiciones: El sistema debe mostrar los gráficos estadísticos atendiendo a la información almacenada en la base de datos.	

<p>Pasos:</p> <ol style="list-style-type: none"> 1- El administrador selecciona la segunda interfaz del componente denominada gráficos. 2- Una vez seleccionada, se mostrarán los gráficos relacionados con información relevante, almacenada en la base de datos.
<p>Resultados esperados: El correcto funcionamiento de los gráficos, mostrando información confiable.</p>
<p>Evaluación: Prueba satisfactoria.</p>
<p>Observaciones: Cumple con la precondition establecida de mostrar los gráficos estadísticos atendiendo a la información almacenada en la base de datos.</p>

Tabla 18: Pruebas de aceptación [RF5].

Iteración No.: 1	
<p>Nombre del requisito a probar: Visualizado solo por los administradores del portal.</p>	<p>Número del requisito a probar: [RF6.]</p>
<p>Descripción del requisito a probar: El componente solo se debe visualizar desde el panel de control al cual solo tienen acceso los administradores del portal.</p>	
<p>Precondiciones: El componente solo se puede tener acceso desde el panel de control.</p>	
<p>Pasos:</p> <ol style="list-style-type: none"> 1- Autenticarse como administrador del portal. 2- Una vez autenticado como administrador, dirigirse al panel de control y seleccionar <i>Event Trace Portlet</i>. 3- Se muestra el componente al administrador con todas sus funcionalidades. 	
<p>Resultados esperados: Solo tengan acceso al componente los administradores del portal.</p>	
<p>Evaluación: Prueba satisfactoria.</p>	
<p>Observaciones: Cumple con la precondition establecida de que al componente solo se puede tener acceso desde el panel de control.</p>	

Tabla 19: Pruebas de aceptación [RF6].

3.4 Pruebas de estrés

Las pruebas de estrés se les realizan a los sistemas atendiendo a las demandas y sus respuestas, no se consideran pruebas funcionales. Estas se realizan buscando la posibilidad de posibles fallos en cuanto a:

- Límites de conexiones que soporta la aplicación.
- Reducción de las caídas del sistema.

- Permitir tomar decisiones acerca de configuraciones de *hardware*, ajustes de *software* y selección de arquitecturas.

Summary Report

Nombre: Summary Report

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Display Only: Escribir en Log Sólo Errores Successes

Label	# Muestras	Media	Mín	Máx	Std. Dev.	% Error	Rendimiento	Kb/sec	Avg. Bytes
Portlet	50	4117	3129	5023	420.78	0.00%	9.3/sec	444.29	48943.9
TOTAL	50	4117	3129	5023	420.78	0.00%	9.3/sec	444.29	48943.9

Ilustración 10: Prueba para 50 usuarios utilizando jmeter 2.3.1. Elaboración propia.

Prestaciones del *Hardware* utilizado:

- Processor: Intel(R) Core i5(MT) i5 -5200 CPU @2.20GHZ @2.20GHZ.
- Memoria RAM: 8GB DDR3 1600MHZ.

Se tuvo en cuenta factores críticos como el tiempo de respuesta del componente 9.3/sec y la cantidad de usuarios conectados al mismo tiempo 50. De acuerdo a estos resultados, las pruebas se realizaron satisfactoriamente obteniéndose valores esperados con el tiempo de respuesta que debe de poseer el componente para su correcto funcionamiento en el Liferay Portal.

3.5 Conclusiones parciales

En el capítulo se explicaron los procesos que intervinieron en la implementación de la solución propuesta. Además, los métodos y pruebas que se le aplicaron al *software* para lograr mejoras sustanciales en las capas de lógica del negocio y presentación, generando a su vez una interfaz más amigable e intuitiva para el usuario.

- Se describió el estándar de código con el que se realizó la implementación del componente.
- Se realizaron las estrategias de pruebas para aplicarle al *portlet* las pruebas de caja negra.
- Se realizaron pruebas de aceptación al componente para ver la certificación por parte del cliente.
- Para poner a prueba el rendimiento del componente se le realizaron las pruebas de estrés.

CONCLUSIONES

- La herramientas y tecnologías seleccionadas para la elaboración del *portlet* de gestión de tazas para el Liferay Portal (CE) facilitaron el desarrollo fluido y sencillo de una aplicación robusta con un entorno bien definido.
- Con el desarrollo del componente de gestión de trazas y eventos del Liferay Portal se logró monitorear, estudiar y evaluar el comportamiento de los usuarios dentro del sistema para lograr fiabilidad en la toma de decisiones de los administradores de los portales.
- Con el componente *Event Trace Portlet* se evita la inversión de capital para adquirir el componente *Audit Portlet*, desarrollado solo para la versión Liferay Portal (EE).
- El componente desarrollado garantiza la fiabilidad de la información referente a las trazas de los usuarios dentro del sistema. Demostrado a través de las diversas pruebas realizadas al sistema el buen funcionamiento del mismo.

RECOMENDACIONES

- Extender en funcionamiento del componente con el objetivo de capturar nuevos eventos para aumentar la visión de los administradores del portal.

BIBLIOGRAFÍA (referencias)

2.0, C. V. E. *¿Qué es La Web 2.0?* [Página Web]. Comunidad Virtual Educación 2.0, 2011, vol. 2015, Disponible en: <https://lablogtacora.wordpress.com/%C2%BFque-es-la-web-20/>.

ALMIRÓN, C. G. *Introducción a JSF Java | Adictos al Trabajo* [Página Web]. AdictosAlTrabajo.com, [Consultado el: Diciembre de 2015]. Disponible en: <http://www.adictosaltrabajo.com/tutoriales/introduccion-jsf-java/>.

BLANCO-CUARESMA, S. *Metodologías de desarrollo*. [Blog]. 2008, vol. 2015, Disponible en: <http://www.marblestation.com/?p=644>.

BOK, S. M. *Historia de usuario - Scrum Manager BoK* [Página Web]. Scrum Manager Body of Knowledge, [Consultado el: Febrero de 2016]. Disponible en: http://www.scrummanager.net/bok/index.php?title=Historia_de_usuario.

BRIANO, F. *Introducción a Spring Framework Java* [Página Web]. picando codigo, [Consultado el: Diciembre de 2015]. Disponible en: <http://picandocodigo.net/2010/introduccion-a-spring-framework-java/>.

BUITRAGO, J. *¿Qué es Drupal? ¿Para qué se usa?* [Página Web]. Drupal Groups, [Consultado el: Diciembre de 2015]. Disponible en: <https://groups.drupal.org/node/148379>.

CANO, R. B. *BASE DE DATOS DISTRIBUIDAS* [Blog]. UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO, 2007, vol. 2015, Disponible en: <http://rburgoscan.blogspot.com/>.

CELY, C. P. S. *Autenticación de usuarios* [Página Web: Sitio Oficial]. Washington, D.C: Comisión Interamericana de Telecomunicaciones, Organización de los Estados Americanos, [Consultado el: Diciembre de 2015]. Disponible en: http://www.oas.org/en/citel/infocitel/2006/junio/seguridad_e.asp.

CORONADO, A. *Desarrollo de Hooks en Liferay « Albert Coronado* [Página Web]. Formación, Consultoría y Full Stack Development (100% Freelance) [Consultado el: Diciembre de 2015]. Disponible en: <http://www.albertcoronado.com/2011/01/28/desarrollo-de-hooks-en-liferay/>.

DÍAZ, A. J. P. *Tomcat, Apache Tomcat, Jakarta Tomcat* [Página Web]. ajpdsoft.com, [Consultado el: Diciembre de 2015]. Disponible en: <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=769>.

EMANUEL. *Estandares de Código* [Página Web]. Slideshare, [Consultado el: Marzo de 2016]. Disponible en: <http://es.slideshare.net/PiXeL16/estandares-de-codigo-emanuel>.

GALVIS, A. F. P.; SANCHEZ, J. A., et al. *Auditoría de sistemas de información* [Página Web]. INSTITUCIÓN UNIVERSITARIA DE ENVIGADO FACULTAD DE CIENCIAS

- EMPRESARIALES CONTADURÍA PÚBLICA, [Consultado el: Febrero de 2016]. Disponible en: <http://www.gerencie.com/auditoria-de-sistemas-de-informacion.html>.
- GARCÍA, A. P. *Maven, nunca antes resultó tan fácil compilar, empaquetar, ...* | *adictosaltrabajo* [Página Web]. AdictosAITrabajo.com, [Consultado el: Marzo de 2016]. Disponible en: <https://www.adictosaltrabajo.com/tutoriales/maven/>.
- GARTNER. *Liferay reconocido como Líder en Portales por Sexto Año* [Página Web]. gartnet.com liferay.com, [Consultado el: Diciembre de 2015]. Disponible en: https://www.liferay.com/es/about-us/awards/gartnermq-portals?tactic_id=59223882.
- GENBETA.DEV. *Eclipse IDE* [Página Web]. GenBeta.Dev, [Consultado el: Diciembre de 2015]. Disponible en: <http://www.genbetadev.com/herramientas/eclipse-ide>.
- INFORMÁTICA, A. Y. S. *Auditoría y Seguridad Informática, Cuidando Tu Negocio* [Página Web]. ASI, Auditoría y Seguridad Informática SA, [Consultado el: Diciembre de 2015]. Disponible en: <http://www.auditoria.com.mx/auditoria-de-sistemas-de-informacion>.
- ING. JAVIER A. VOOS, I. E. G., ING. FERNANDO CAGNOLO. Portal de Aplicaciones Médicas. 2012, n^o p. 11. Disponible en: <http://www.bioingenieria.edu.ar/grupos/geic/biblioteca/Trabypres/T03TCAr10.pdf>.
- INSPIRAIT, S. I. E. T. *Portales Empresariales* [Página Web]. Inspirait, Soluciones inspiradas en ti, [Consultado el: Diciembre de 2015]. Disponible en: <http://www.inspirait.com/portales-empresariales/>.
- LAPUENTE, M. J. L. *Hipertexto, el nuevo concepto de documento en la cultura de la imagen*. Doctorado, Biblioteconomía y Documentación. Universidad Complutense de Madrid, 2013.
- LEBRÚN, C. A. V. *INTEGRACIÓN DE HERRAMIENTAS DE TECNOLOGÍAS DE INFORMACIÓN "PORTALES COLABORATIVOS DE TRABAJO" COMO SOPORTE EN LA ADMINISTRACIÓN DEL CONOCIMIENTO*. Tutor: Carballido, J. M. Doctoral, Universidad Cristóbal Colón 2005.
- LEON, J. D. *Portales y portlets web* [Página Web]. SilderShare: es.sildershare.net, [Consultado el: Marzo de 2016]. Disponible en: <http://es.slideshare.net/jossydeleon/portales-y-portlets-web-9377907>.
- LERMA, E. V. *Introducción a Primefaces* [Página Web]. AdictosAITrabajo.com: [Consultado el: Enero de 2016]. Disponible en: <https://www.adictosaltrabajo.com/tutoriales/introduccion-primefaces/>.

- LIC. ERVIN FLORES, J. L. C. *AUP Ingeniería de Software* UNIVERSIDAD UNION BOLIVARIANA, CARRERA DE INGENIERIA DE SISTEMAS, [Consultado el: Diciembre de 2015]. Disponible en: http://ingenieriadesoftware.mex.tl/63758_AUP.html.
- . *METODOLOGÍAS ÁGILES, PROCESO UNIFICADO ÁGIL (AUP)*. [Ingeniería de Software]. Universidad Union Bolivariana, publicado el: Diciembre de última actualización: Diciembre. Disponible en: http://www.academia.edu/7894130/METODOLOGIAS_AGILES_AUP.
- LIFERAY. *Liferay Portal 6.2 Improves Mobility and WEM | CMS Report* [Página Web]. liferay.com, [Consultado el: Enero de 2016]. Disponible en: http://www.ite.educacion.es/formacion/materiales/155/cd/modulo_1_Iniciacionblog/concepto_de_web_20.html.
- . Liferay Portal CE & Liferay Portal EE. 2012, nº p. 2. [Consultado el: 2016]. ISSN 083011.
- LIFERAY, D. N. *What is Service Builder? - Liferay 6.2 - Liferay Developer Network* [Página Web]. dev.liferay.com: Liferay Developer, [Consultado el: Marzo de 2016]. Disponible en: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-2/what-is-service-builder.
- MAYOR, A. C. *CMS, LMS y LCMS. Definición y diferencias* [Página Web]. Centro de Comunicación y Pedagogía, [Consultado el: Diciembre de 2015]. Disponible en: <http://www.centrocp.com/cms-lms-y-lcms-definicion-y-diferencias/>.
- MIFSUD, E. *MONOGRÁFICO: Introducción a la seguridad informática - Mecanismos básicos de seguridad* [Página Web]. España: Observatorio Tecnológico, Gobierno de España, [Consultado el: Diciembre de 2015]. Disponible en: <http://recursostic.educacion.es/observatorio/web/es/component/content/article/1040-introduccion-a-la-seguridad-informatica?start=2>.
- PROFESORADO, I. N. D. T. E. Y. D. F. D. *Concepto de Web 2.0 | Multimedia y Web 2.0* Instituto Nacional de tecnología Eduactivas y de Formación del Profesorado, [Consultado el: Enero de 2016]. Disponible en: http://www.ite.educacion.es/formacion/materiales/155/cd/modulo_1_Iniciacionblog/concepto_de_web_20.html.
- RANCEL, M. R. *¿Qué es y para qué sirve Joomla? CMS potente e interesante para crear páginas web de todo tipo (CU00403A)* [Página Web]. aprenderaprogramar.com, Didáctica y Divulgación de la programación, [Consultado el: Diciembre de 2015]. Disponible en: http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=119:ig-ue-es-y-para-que-sirve-joomla-cms-potente-e-interesante-para-crear-paginas-web-de-todo-tipo-cu00403a&catid=38:curso-qcreacion-web-con-joomla-desde-ceroq&Itemid=152.
- RICHARD SEVOZ, J., BRIAN KIM. *Liferay in Action*. 1 ed. Científico-Técnico, 2012. 378 p. ISBN L-935182-82-x.

ROJAS, M. A. y GARCÍA, J. C. M. INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES. Marzo 2004, nº p. 12.

ROLDÁN, C. S. *¿Qué es una Auditoría Informática?* [Página Web]. Estados Unidos: CODEJOBS, [Consultado el: Diciembre de 2015]. Disponible en: <https://www.codejobs.biz/es/blog/2013/02/25/que-es-una-auditoria-informatica>.

ROSAS, J. E. S. *Liferay: Portales web para el trabajo colaborativo de empresas*. [Blog]. Antartec, 2010, vol. 2015, Disponible en: <http://blogs.antartec.com/opensource/2010/04/liferay-portal/>.

SCAMARIO, F. *Instalación de Liferay*. [Blog]. WordPress, 2010, vol. 2015, Disponible en: <http://francescoscamarcio.com/2010/12/13/instalacion-de-liferay/>.

TECNOLÓGICAS, S. S. *¿Qué es OpenCms?* [Página Web]. Saga Soluciones Tecnológicas, [Consultado el: Diciembre de 2016]. Disponible en: <http://www.templatesaga.com/nav/que-es-opencms/>.

ANEXOS

Figure 1. Magic Quadrant for Horizontal Portals



Source: Gartner (September 2015)

Ilustración 11: Cuadrante mágico de Gartner (GARTNER, 2015).

Control Panel | Users | Sites | Apps | Configuration | My Sites | 0 | Test Test

Users and Organizations | User Groups | Roles | Password Policies | Monitoring | **Event Trace Portlet**

Registros | Gráficos

Buscar:

(1 of 6) | 1 | 2 | 3 | 4 | 5 | 6 | 10

Usuario	Evento	Dominio	Dirección IP	Descripción	Fecha	Hora
<input type="text"/>	Seleccione	<input type="text"/>	<input type="text"/>	Seleccione	Inicial Final	Inicial Final
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	21:40
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	21:44
Invitado	Change Page	localhost	127.0.0.1	Entró al sistema	26/05/2016	21:44
Invitado	Change Page	localhost	127.0.0.1	Salió del sistema	26/05/2016	21:44
Invitado	Change Page	localhost	127.0.0.1	Welcome	26/05/2016	21:44
Invitado	Change Page	localhost	127.0.0.1	filtro fecha	26/05/2016	21:46
Invitado	Change Page	localhost	127.0.0.1	Control Panel	26/05/2016	21:46
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	21:46
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	21:50
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	21:50
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	22:01
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	22:01

(1 of 6) | 1 | 2 | 3 | 4 | 5 | 6 | 10

Ilustración 12: Interfaz de filtrado por la descripción del evento. Elaboración propia.

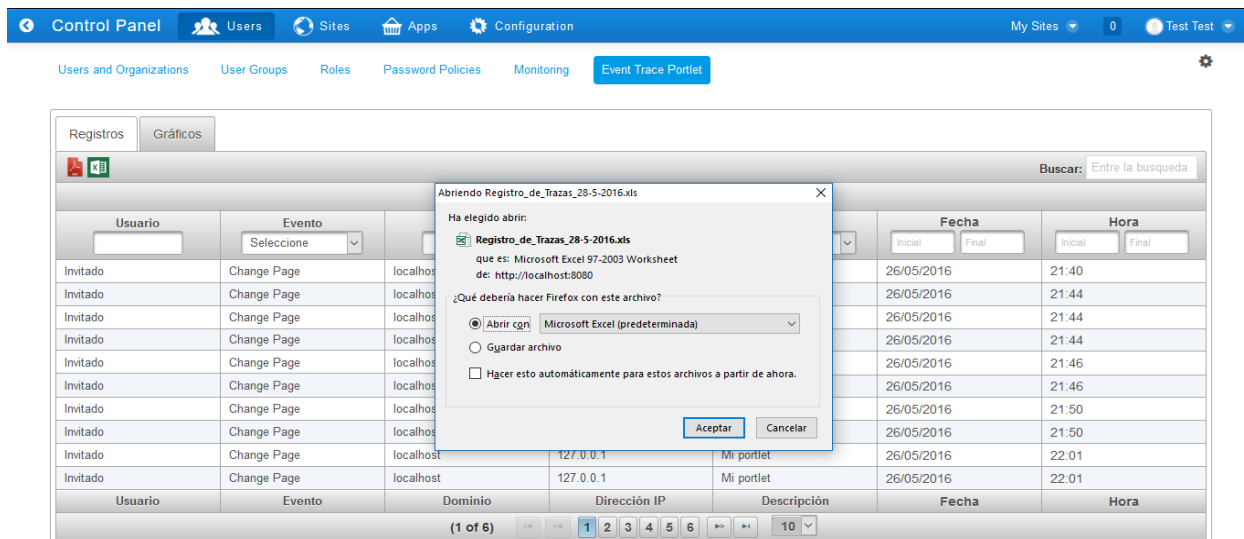


Ilustración 13: Ventana emergente de descarga del documento a formato excel. Elaboración propia.

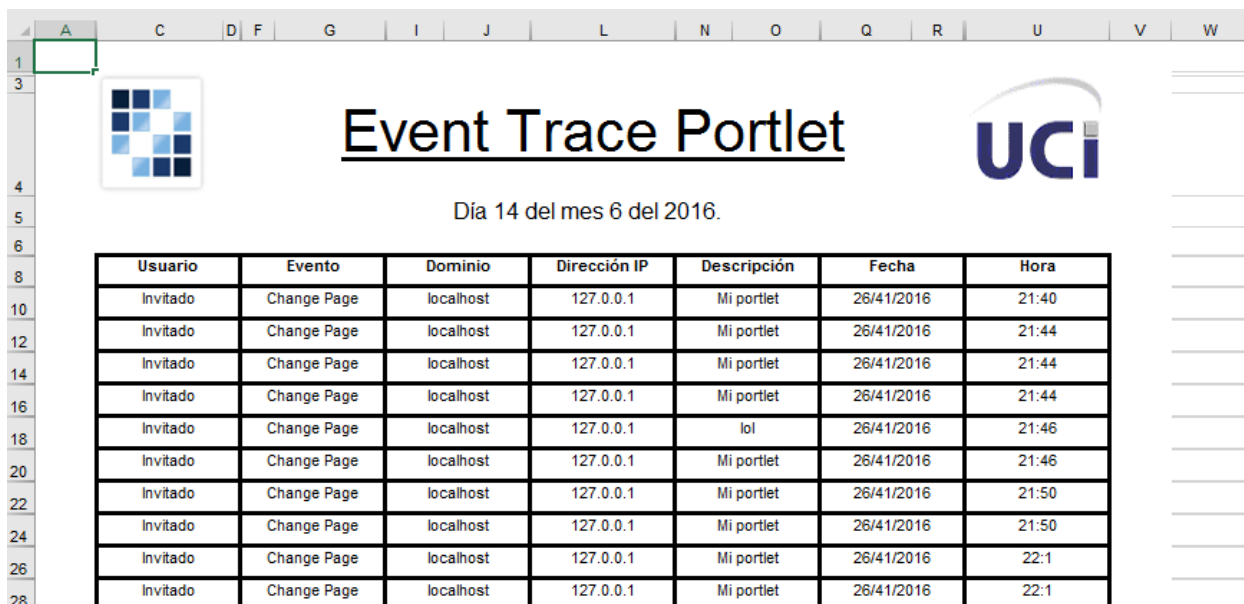


Ilustración 14: Reporte a formato excel. Elaboración propia.

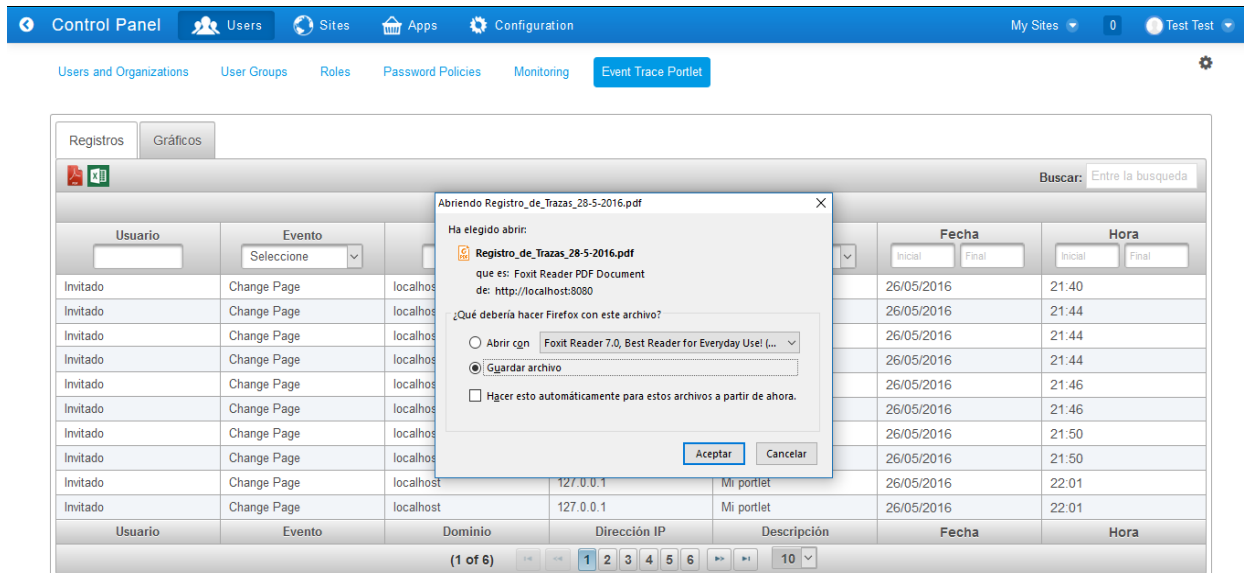


Ilustración 15: Ventana emergente de descarga del documento a formato pdf. Elaboración propia.

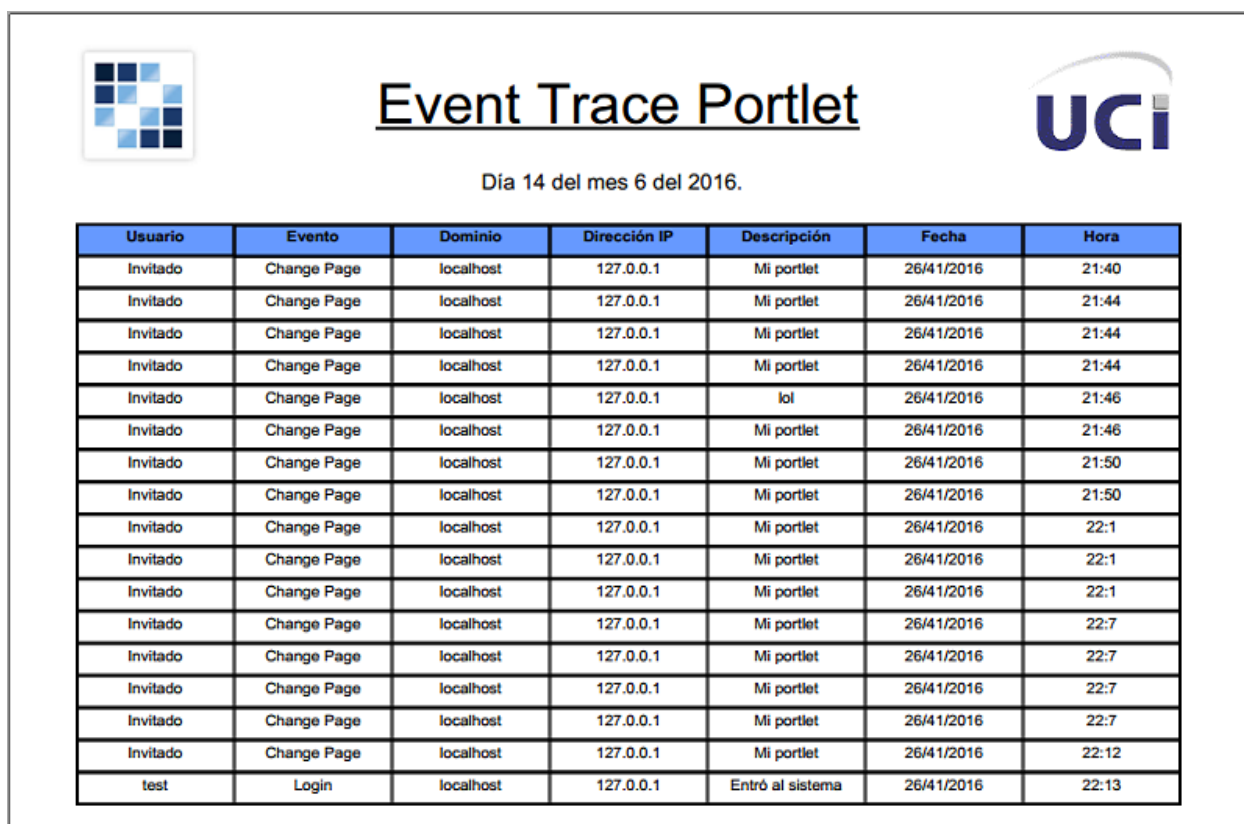


Ilustración 16: Reporte a formato pdf. Elaboración propia.

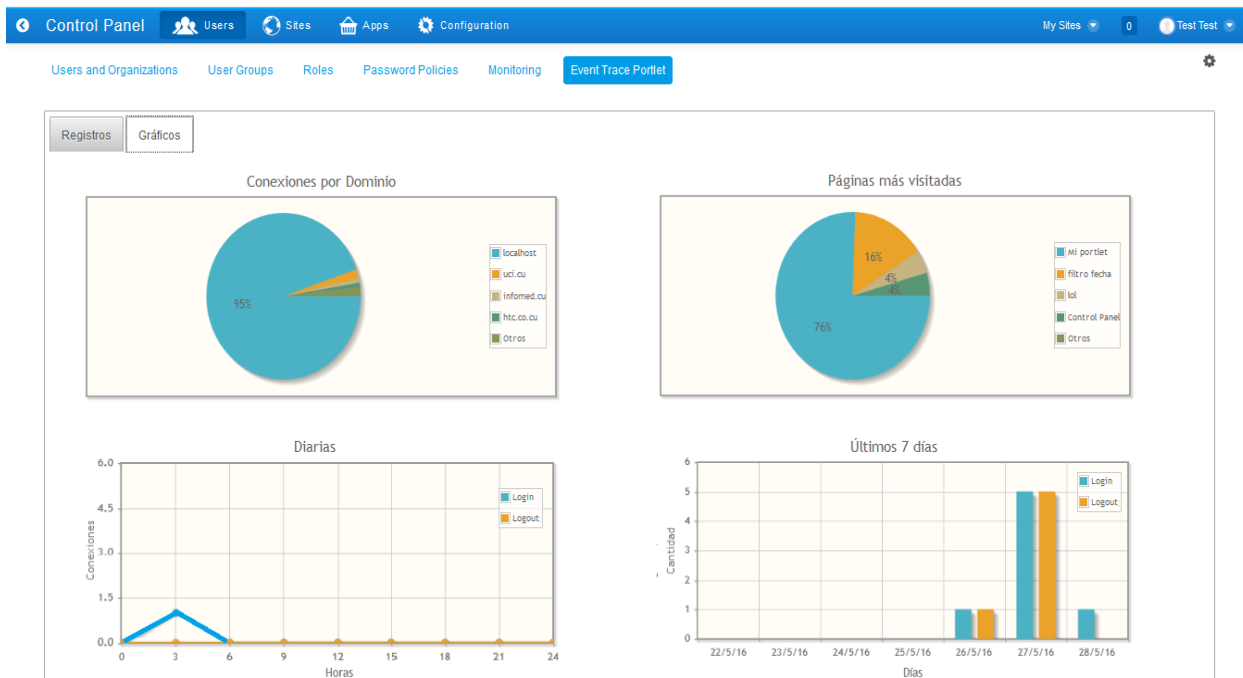


Ilustración 17: Interfaz del tablero de control. Elaboración propia.

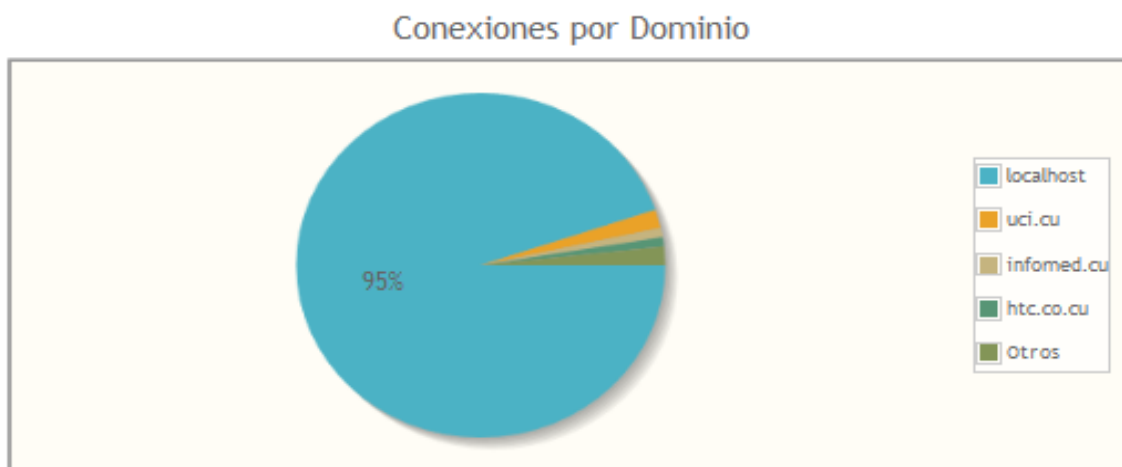


Ilustración 18: Gráfico de conexiones por dominio. Elaboración propia.



Ilustración 19: Gráfico de páginas más visitadas. Elaboración propia.

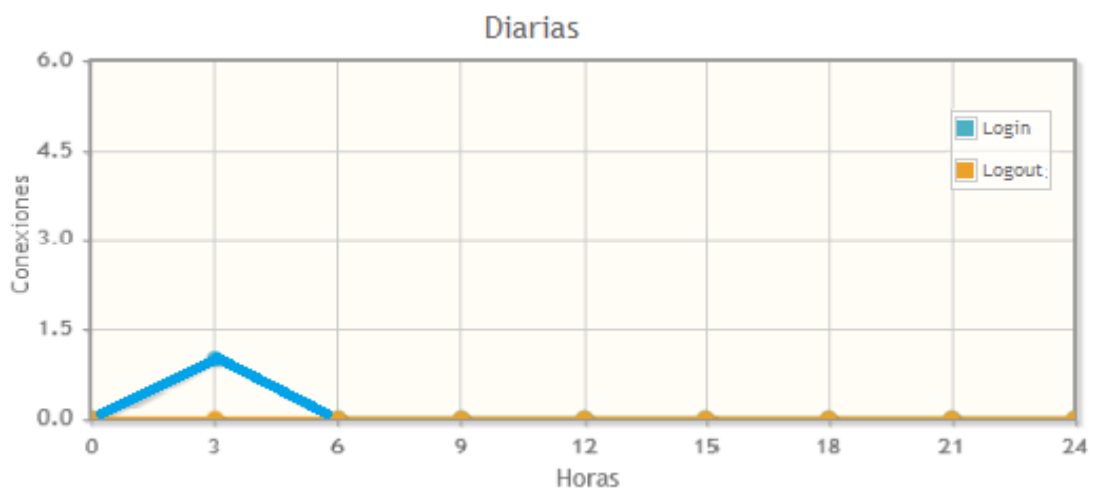


Ilustración 20: Gráfico de conexiones diarias. Elaboración propia.

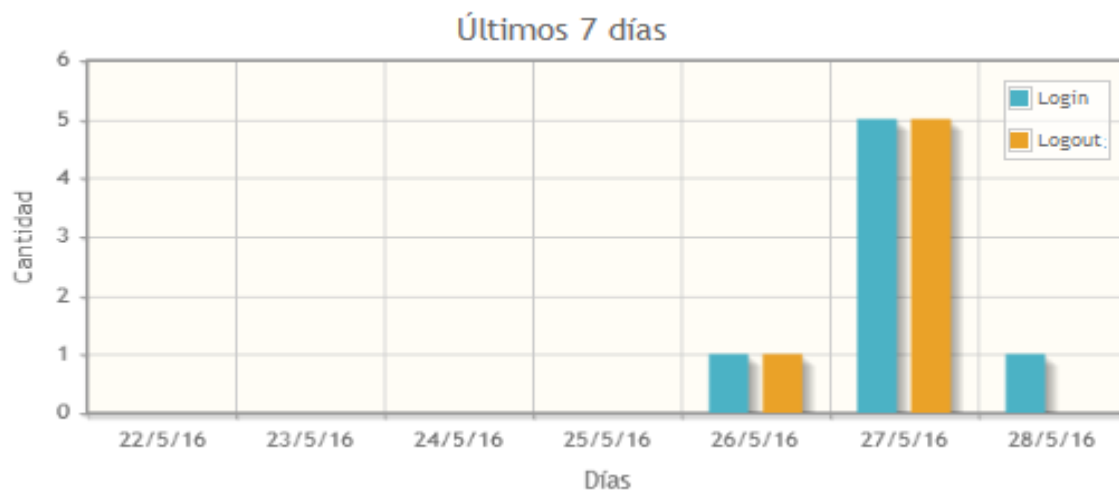


Ilustración 21: Gráfico de conexiones en los últimos 7 días. Elaboración propia.

Control Panel | Users | Sites | Apps | Configuration | My Sites | 0 | Test Test

Users and Organizations | User Groups | Roles | Password Policies | Monitoring | **Event Trace Portlet**

Registros | Gráficos

Buscar:

(1 of 6) 1 2 3 4 5 6 10

Usuario	Evento	Dominio	Dirección IP	Descripción	Fecha	Hora																																															
<input type="text"/>	Seleccione	<input type="text"/>	<input type="text"/>	Seleccione	inicial Final	inicial Final																																															
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	<div style="border: 1px solid gray; padding: 2px;"> <p>May 2016</p> <table border="1"> <tr><td>S</td><td>M</td><td>T</td><td>W</td><td>T</td><td>F</td><td>S</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td></td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr> <tr><td></td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr> <tr><td></td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr> <tr><td></td><td>29</td><td>30</td><td>31</td><td></td><td></td><td></td><td></td></tr> </table> </div>		S	M	T	W	T	F	S		1	2	3	4	5	6	7		8	9	10	11	12	13	14		15	16	17	18	19	20	21		22	23	24	25	26	27	28		29	30	31				
S	M	T	W	T	F	S																																															
	1	2	3	4	5	6	7																																														
	8	9	10	11	12	13	14																																														
	15	16	17	18	19	20	21																																														
	22	23	24	25	26	27	28																																														
	29	30	31																																																		
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	22:01																																															
Invitado	Change Page	localhost	127.0.0.1	Mi portlet	26/05/2016	22:01																																															

(1 of 6) 1 2 3 4 5 6 10

Ilustración 22: Interfaz de filtrado por fecha. Elaboración propia.

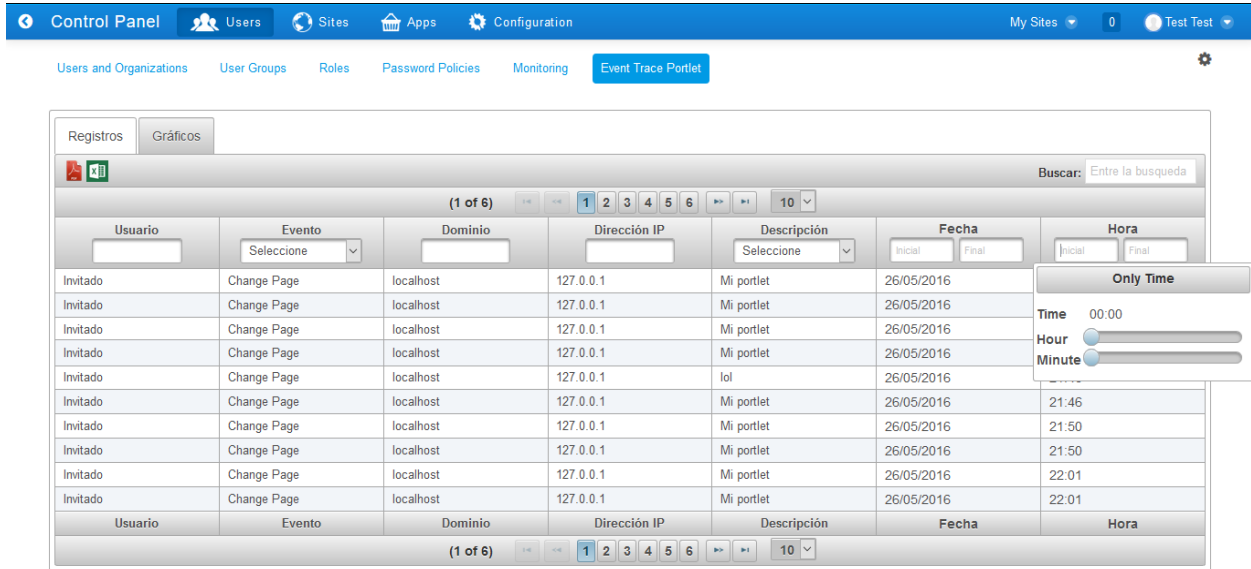


Ilustración 23: Interfaz de filtrado por hora. Elaboración propia.

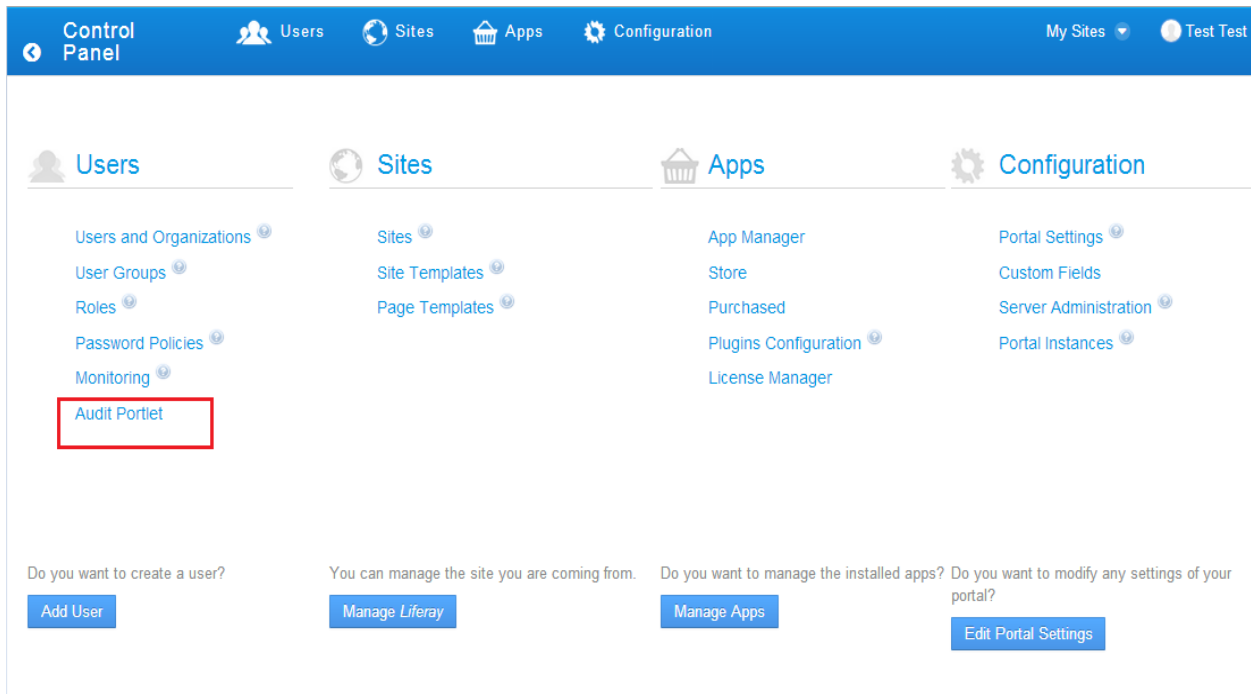


Ilustración 24: El componente Audit Portlet del Liferay Portal (EE) en el panel de control (LIFERAY, 2014a).

The screenshot shows the Liferay Portal Control Panel with a blue header. The header contains navigation links: Control Panel, Users, Sites, Apps, and Configuration. On the right, it shows 'My Sites' with a dropdown arrow, a '0' indicator, and 'Test Test' with a dropdown arrow. Below the header, there are four main sections: Users, Sites, Apps, and Configuration. Each section has a list of sub-items. In the 'Users' section, 'Event Trace Portlet' is highlighted with a red rectangular box. Below the sections, there are four prompts with corresponding buttons: 'Do you want to create a user?' with 'Add User', 'You can manage the site you are coming from.' with 'Manage Liferay', 'Do you want to manage the installed apps?' with 'Manage Apps', and 'Do you want to modify any settings of your portal?' with 'Edit Portal Settings'.

Ilustración 25: El componente Event Trace Portlet del Liferay Portal (CE) en el panel de control. Elaboración propia.

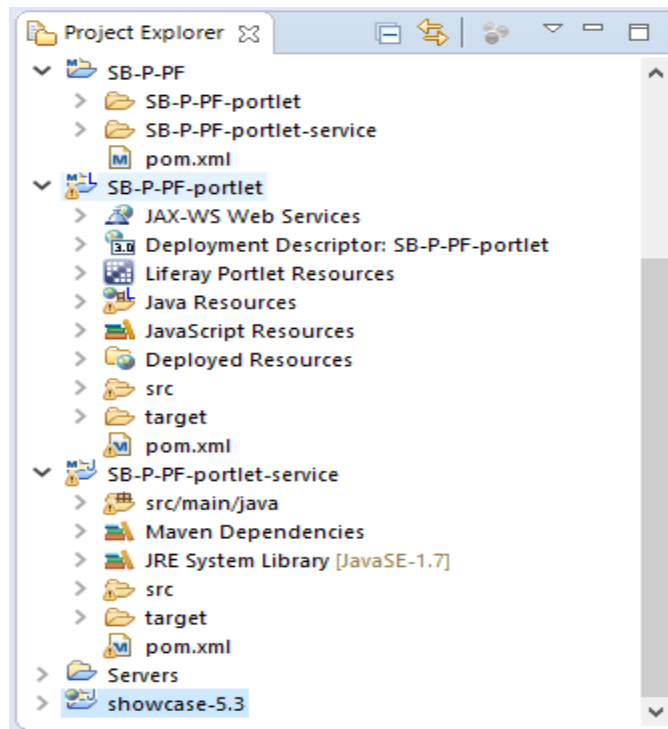


Ilustración 26: Estructura del proyecto generado con maven. Elaboración propia.

Control Panel Users Sites Apps Configuration My Sites Test Test

Portal Settings Custom Fields Server Administration Portal Instances Workflow Audit Reports

Keywords Search

Page 1 of 2 20 Items per Page Showing 1 - 20 of 33 results. - First Previous Next Last ->

User ID	User Name	Resource ID	Resource Name	Resource Action	Client IP	Create Date
10198	Test Test	10175	Page	View	0:0:0:0:0:0:1	2 Seconds Ago
10198	Test Test	10198	User	Login	0:0:0:0:0:0:1	4 Seconds Ago
10198	Test Test	10184	Page	View	0:0:0:0:0:0:1	4 Seconds Ago
10528	Joe Bloggs	10528	User	Logout	0:0:0:0:0:0:1	9 Seconds Ago
10528	Joe Bloggs	10198	User	Update	0:0:0:0:0:0:1	15 Seconds Ago
10528	Joe Bloggs	10528	User	Unassign	0:0:0:0:0:0:1	15 Seconds Ago
10528	Joe Bloggs	10528	User	Unassign	0:0:0:0:0:0:1	15 Seconds Ago
10528	Joe Bloggs	10528	User	Assign	0:0:0:0:0:0:1	15 Seconds Ago
10528	Joe Bloggs	10528	User	Assign	0:0:0:0:0:0:1	15 Seconds Ago
10528	Joe Bloggs	10175	Page	View	0:0:0:0:0:0:1	26 Seconds Ago
10528	Joe Bloggs	10544	Page	View	0:0:0:0:0:0:1	32 Seconds Ago
10528	Joe Bloggs	10184	Page	View	0:0:0:0:0:0:1	35 Seconds Ago
10528	Joe Bloggs	10583	Page	View	0:0:0:0:0:0:1	36 Seconds Ago
10528	Joe Bloggs	10184	Page	View	0:0:0:0:0:0:1	38 Seconds Ago
10528	Joe Bloggs	10528	User	Login	0:0:0:0:0:0:1	39 Seconds Ago

Ilustración 27: Interfaz del componente Audit Portlet del Liferay Portal (EE) (LIFERAY, 2014a).

Id del escenario	Escenario	Variable 1 Nombre	Respuesta del Sistema	Resultado de la Prueba
EC 1	Registrar eventos de los usuarios.	V Se lanza un evento del <i>core</i> del Liferay Portal (CE) de los cuales se monitorean con los <i>hook</i> .	Adicionar una traza a la base de datos.	Se agrega una acción al sistema.
		NA No se lanza un evento del <i>core</i> del Liferay Portal (CE) de los cuales se monitorean con los <i>hook</i>		
EC 2	Obtener listado de los eventos de los usuarios.	V Cargar desde el panel de control el componente de gestión de trazas.	Mostrar una lista de todas las trazas almacenadas en la base de datos.	Se agrega una acción al sistema.
		NA No se cargar el componente de gestión de trazas.		

EC 4.1	Reporte (exportar) a formato pdf.	V Seleccionar el icono de pdf en la interfaz del <i>portlet</i> .	Mostrar una ventana emergente con las opciones de salvar o visualizar el documento generado.	Se agrega una acción al sistema.
		NA No se selecciona el icono de pdf en la interfaz del <i>portlet</i> .		
EC 4.2	Reporte (exportar) a formato excel.	V Seleccionar el icono de excel en la interfaz del <i>portlet</i> .	Mostrar una ventana emergente con las opciones de salvar o visualizar el documento generado.	Se agrega una acción al sistema.
		NA No se selecciona el icono de excel en la interfaz del <i>portlet</i> .		

EC 5	Tablero de control para visualizar el comportamiento de los temas analizados.	V Seleccionar el <i>tab</i> Gráficos en la interfaz principal del <i>portlet</i> .	Mostrar en la vista el comportamiento de los temas analizados en gráficos estadísticos de pastel , barras y líneas.	Se agrega una acción al sistema.
		NA No se selecciona el <i>tab</i> Gráficos.		
EC 6	Visualizado solo por los administradores del portal.	V Al identificarse un usuario como administrador del sistema.	Mostrar en el panel de control el componente de gestión de trazas.	Se agrega una acción al sistema.
		NA El usuario identificado no es administrador del sistema.	No se muestra el componente en el panel de control del usuario identificado.	

Tabla 20: Escenarios de prueba (continuación).