UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9

Implementación de los módulos de RRHH y AR del portal Calidad de la Facultad 9.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

AUTOR(ES):

Yosniel Gongora Benitez

TUTOR(ES):

Ing. Yoandry Lazo Nodarse

Ing. Yudiel Rodríguez Larrazabal

Ciudad de la Habana, 2010.

Año 52 de la Revolución

" ... Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes."

Fidel Castro.

"Ayudar al que lo necesita no solo es parte del deber, sino de la felicidad."

José Martí.

DEDICATORIA

Esta tesis está dedicada a mi madre, quién me trajo a este mundo y quién ha sido mi mayor inspiración en estos 18 años de estudio, que ha hecho lo posible y lo imposible para darme lo necesario y para que pueda estudiar sin ninguna otra preocupación, a ella que me ha brindado su amor y apoyo incondicional. A mi abuela que la quiero muchísimo y que también ha sido parte de mi inspiración y me ha apoyado todos estos años. A mi padrastro, que más bien ha sido un padre para mí y me ha guiado todo este tiempo. A mi hermanastra, que la quiero como una hermana. A mi abuelo, mis tíos y tías que siempre se han preocupado por mí y por mis estudios y que no han dudado en ayudarme, a ellos por estar presente en todo momento. A mis primos y primas que son mis tesoros, así como aquellos que no llevan mi sangre y para mí son parte de la familia, porque los quiero como tíos, padres o hermanos, a toda mi familia, a mis compañeros de estudio, a todos los que han formado parte de mi formación, a todos ustedes va dedicada mi tesis.

AGRADECIMIENTOS

Agradezco a la Revolución y a Fidel por darle a cada joven la oportunidad de estudiar, de lograr sus sueños. A mi madre, por todo su esfuerzo en estos 18 años de estudio, que ha hecho lo posible y lo imposible para darme lo necesario y para que pueda estudiar sin ninguna otra preocupación, a ella que me ha brindado su amor y apoyo incondicional. A mi abuela que la quiero muchísimo y que también ha sido parte de mi inspiración y me ha apoyado todos estos años. A mi padrastro, que ha sido un padre para mí y me ha guiado todo este tiempo. A mi hermanastra, que la quiero como una hermana. A mi abuelo, mis tíos y tías que siempre se han preocupado por mí y por mis estudios y que no han dudado en ayudarme, a ellos por estar presente en todo momento. A mis primos y primas que son mis tesoros, así como aquellos que no llevan mi sangre y para mí son parte de la familia, porque los quiero como tíos, padres o hermanos, a toda mi familia, a mis compañeros de estudio, a mis tutores por el gran trabajo que han realizado, por el apoyo que me dieron, por demostrarme en todo momento que este sueño era posible y por la preocupación mostrada en todo el trayecto de la tesis, a mi tribunal que me ha aconsejado en cada corte, y que me llevaron a mejorar cada día, a mi compañero Félix, por todo el apoyo y la ayuda que me brindó, a Mailín que me solucionó muchas veces los problemas con el Word, a mis compañeros aquí en la UCI, tanto a los del primer grupo, como los del actual, a los que no forman parte de estos grupos, pero en un momento u otro hemos compartido momentos buenos y malos, a todos los profesores que han tenido que ver con mi formación, a todos ellos agradecer porque han contribuido a que este hoy aquí, graduándome de ingeniero en ciencias informáticas.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2010.

Yosniel Gongora Benitez

_____

Yoandry Lazo Nodarse

Yudiel Rodríguez Larrazabal

_____

DATOS DE CONTACTO

<Insertar breve currículum e información de contacto del tutor>

<Insertar breve currículum e información de contacto del co-tutor>

<Insertar breve currículum e información de contacto del consultor>

<Insertar breve currículum e información de contacto del asesor>

OPINIONES Y AVALES

<En este acápite se ponen en hojas por separado todos los avales (si el trabajo los tiene) de participación en eventos científicos o avales por empresas o centros de investigación o estudio, acreditados en certificación. También se presentan en este acápite todas las opiniones de los beneficiarios de los resultados de trabajo científico>

OPINIÓN DEL TUTOR

<En este acápite se incluye la opinión del tutor del trabajo de diploma>

RESUMEN

En la Universidad de Ciencias Informáticas (UCI) se construyen aplicaciones para distintas esferas, por lo que existen grupos para controlar o realizar la gestión de la calidad de estas aplicaciones. Pero debido al difícil acceso a algunos recursos en la Red Global y las restricciones impuestas a determinados servicios, se dificulta la búsqueda de información segura y confiable. A lo mencionado anteriormente se suma el hecho de que no se cuenta con un lugar donde los usuarios puedan encontrar la información que necesitan, o se desconoce del mismo, causando desconocimiento en las personas que realizan las tareas relacionadas con la calidad e imposibilitando un mejor desempeño en las áreas de trabajo tanto a nivel nacional como en el centro. En el Grupo de Calidad de la Facultad 9 se detectó que no se realiza una buena gestión de los datos generados en las actividades realizadas, que existe retraso en el cumplimiento de las actividades, que hay poco control de las tareas que se desarrollan y poca interacción entre el personal interesado en aprender y el personal con experiencia, por lo que se hace necesario gestionar toda la información obtenida como producto de las acciones que se ejecutan en el grupo y brindar otros servicios que ayudarán a mejorar en estos aspectos.

Dada la anterior problemática se identificó el siguiente **problema a resolver**: difícil acceso a la información que se necesita para trabajar, retraso y escaso control detectado en la Gestión de los Recursos Humanos, y de las Auditorías y Revisiones en el Grupo de Calidad de la Facultad 9.

Para dar solución al problema planteado se trazó como **objetivo general:** implementar los módulos de Auditoría y Revisiones y el módulo de Gestión de Recursos Humanos del Portal del grupo de Calidad de la Facultad 9, para agilizar las actividades del grupo y tener un mejor control de todas las acciones que se desarrollan.

Para cumplir con lo planteado en los objetivos, se utilizaron los materiales y métodos que más se acoplaban a las necesidades o exigencias planteadas por el cliente, obteniéndose una aplicación acorde a los requisitos especificados. La aplicación fue probada para comprobar la ausencia de errores o funciones no existentes, mediante la prueba de caja negra, específicamente casos de pruebas, los cuales demostraron que el producto, o sea el portal, es completamente funcional y está listo para ser usado.

PALABRAS CLAVES

Aplicaciones, calidad, implementación, módulos, portal.

ÍNDICES DE TABLAS Y FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE

ÍNDICE

ÍNDICE

INTRODUCCIÓN

Desde los inicios de la humanidad el hombre ha vivido de su trabajo, desarrollando productos para sus beneficios, los que más tarde pasaron de ser para el uso propio a un objeto de comercio. Fueron aumentando los productores y a su vez los productos, por lo que el productor tuvo que buscar la forma de mejorarlos para que el suyo fuera mejor, dando inicio a lo que se conoce como competencia. Dado esto se trazó alternativas para obtener buenos resultados, de esta manera va tomando forma lo que se conoce hoy como Calidad, aplicada en disímiles ámbitos pero con semejantes propósitos y objetivos.

Según un artículo del Profesor Doctor Ingeniero Arturo Luis Romero el concepto de calidad viene del latín Qualitas y está asociado a la propiedad o atributo que distingue a las personas, bienes o servicios. (1)

Romero plantea que algunos autores se refieren a calidad de la siguiente forma:

“La calidad es el nivel de excelencia que la empresa ha escogido alcanzar para satisfacer a su clientela clave”. (1)

Concepto de Gestión de Calidad expresado por la ISO: 2000

“actividades coordinadas para dirigir y controlar una organización en lo relativo a la calidad”. (1)

Este se basa en: planificación, control, aseguramiento, mejoramiento continuo de la actividad, quedando definido el Sistema de Gestión de Calidad como:

“sistema de gestión para dirigir y controlar una organización con respecto a la calidad”. (1)

Gloria Nistal Rosique (Socia de la Asociación de Técnicos en Informática, Responsable del Grupo de Trabajo de Calidad del Software del Capítulo de Madrid de esta revista) en un documento suyo plantea que la calidad del software consiste en crear productos según las normas, que satisfagan las necesidades del cliente o usuario, o sea que cumplan con los requisitos y tengan cero defectos. Agrega que un buen software debe tener la capacidad de ser reutilizable. También añade que no basta con hablar de calidad, sino que es necesario aplicarla, y cita una frase, según la autora perteneciente a Crosby que dice:

” podemos decir que la Calidad del Software tiene mucho en común con el sexo, que todo el mundo habla de él y no todos los que hablan lo practican. ” (2)

El pasado siglo (XX) fue importante en la evolución de la calidad, destacándose cinco grandes figuras, son estos *William Eduards Deming, Joseph M. Juran, Armand V. Feigenbaum, Kaoru Ishikawa y Philip B. Crosby*. Muchos otros han contribuido a la misma, pero ellos fueron quienes más impacto causaron.

INTRODUCCIÓN

Arturo Luis Romero considera que existen cinco etapas fundamentales en la evolución de la calidad en este siglo, la primera comienza a principios de 1900 hasta 1930, con la Revolución Industrial, la cual desde el punto de vista productivo representaba la transformación del trabajo manual por el trabajo mecanizado, en esta etapa surgen los inspectores de calidad. Desde entonces se siguió perfeccionando la calidad en cada etapa, surgiendo nuevos métodos e ideas hasta llegar a la quinta y última, que data desde 1990 hasta la fecha, la cual tiene como característica principal la pérdida del sentido de la antigua distinción entre producto y servicio, ya que lo que existe es el valor total para el cliente, denominándose esta etapa Servicio de Calidad Total.

Un servicio de calidad total es un enfoque organizacional global, que hace de la calidad de los servicios, según lo percibe el cliente, la principal fuerza propulsora del funcionamiento de la empresa. (1) Otro paso importante fue el de octubre de 1946, en Londres, cuando surge la Organización Internacional para Estandarización, en inglés (*International Organitation for Estandardization*), por sus siglas ISO, acuerdo tomado por representantes de 25 países. Está integrada por organismos nacionales de normalización, con representantes de cada país participante, actualmente cuenta con 138 países miembros.

El ingreso de Cuba a esta organización está estrechamente vinculado con la designación de Ernesto (Che) Guevara como Ministro de Industrias, el 23 de febrero de 1961, consciente de las demandas de un proceso de desarrollo económico e impuesto de los avances científico técnicos del mundo industrializado, constituyendo un verdadero testimonio de su integridad y amplia visión económica. (1)

Hoy día se encuentra más vigente que nunca la calidad, dado el desarrollo y constante evolución por el que ha transitado el mundo, las personas (clientes) no se conforman con un resultado mediocre, si no que quieren obtener lo mejor, que sea perfecto, atractivo, confiable, fácil de usar, seguro, entre otras muchas características.

En el mundo del software es vital que los productos sean generados con la mayor calidad posible, por tanto las empresas e instituciones involucradas en esta rama llevan a cabo estrategias y/o metodologías para dar cumplimiento a este reto. Desde el comienzo de la Informática se han desarrollado un conjunto de vías para garantizarlo, por lo que está en nuestras manos aplicarla.

Existen actualmente en internet, Sitios de Organizaciones que se dedican a garantizar Calidad para productos de software, a continuación se mencionan dos ejemplos:

- Dirección General de recursos Humanos. En el cual se ofrecen servicios al cliente para gestionar la calidad, el mismo obtuvo el Premio Internacional OX Calidad Web 2009

http://rechum.sec-sonora.gob.mx/pagina/sitios.php
El sitio tiene como misión administrar los Recursos Humanos de la Secretaría de Educación y Cultura con transparencia, responsabilidad, honestidad y eficiencia, ofreciendo servicios públicos de calidad, impulsando la formación de capital humano, procurando la mejora continua dentro de un Sistema de Gestión (Sonora, México).

- International Organization for Standardization. Este es el diseñador más grande de normas internacionales.
http://www.iso.org/iso/home.htm
ISO (la Organización Internacional para la Estandarización) es el diseñador más grande del mundo y publicador de Normas Internacionales.

En los últimos años Cuba se ha sumado a esta revolución de la construcción del software, creando empresas capaces de llevar a cabo estas tareas tanto para el beneficio del país como para ayudar y comerciar con otros países. Por lo que también se ha visto en la necesidad de tener centro o grupos que gestionen la calidad de estas empresas. Por lo que el Ministerio de la Informática *y las* Comunicaciones creó un Centro Nacional de Calidad de Software (Calisoft), el cual realiza pruebas de liberación de todos los entregables y productos finales, pruebas de aceptación con el cliente y pruebas piloto con el usuario final. Calisoft brinda sus servicios a nivel nacional y en la Universidad de Ciencias Informáticas (UCI).
La UCI, entre otros objetivos tiene el de impulsar el desarrollo del software en el país y graduar personal con avanzados conocimientos de este tema y otros. Existen en esta Universidad, dada la cantidad de proyectos que desarrollan aplicaciones tanto para diferentes sectores dentro del país como para el exterior, portales para asegurar la calidad en los proyectos productivos, el Portal de Calidad de la Facultad 1, el cual brinda las funcionalidades a los involucrados en el proceso de gestión de la información de la calidad y de los proyectos productivos de esta facultad, el Portal del Grupo Calidad de la Facultad 7, para la gestión de la información de este grupo.
En la Facultad 9 se hace necesario realizar un aseguramiento de la calidad en los diferentes proyectos con los que cuenta la misma, para lograrlo se llevan a cabo varias actividades, entre ellas las de Recursos Humanos, la cual controla el personal involucrado en los proyectos, las computadoras y los proyectos existentes, y las de Auditoría y Revisiones, estas se encargan de revisar las planificaciones

correspondientes a diferentes aplicaciones con el objetivo de encontrar errores para que puedan ser corregidos en tiempo.

Debido al bloqueo impuesto hace más de 40 años a Cuba, se hace difícil el acceso a algunos recursos en la Red Global, como también se imponen restricciones a determinados servicios, lo que en ocasiones imposibilita un mejor desempeño en las áreas de trabajo tanto a nivel nacional como en el centro, así como se ha comprobado que existen pocos conocimientos relacionados con el aseguramiento de la calidad de los productos que se desarrollan, por parte de los estudiantes y de los trabajadores que se desenvuelven en este tema, no hay interacción entre aquellos que realizan estas actividades y las personas más experimentadas en cuanto a este ámbito, no se tiene un lugar donde se pueda encontrar documentación necesaria para trabajar o los usuarios no tienen conocimiento sobre su existencia, no se puede controlar y archivar de manera rápida y sin el uso de pocos recursos todo lo realizado para la obtención de este término tan importante en una aplicación, identificándose en la facultad 9 lo siguiente:

- ✓ Retraso en la terminación de las actividades del grupo.
- ✓ Lentitud en la búsqueda de información.
- ✓ Duplicado de información almacenada.
- ✓ Poco conocimiento por parte de los estudiantes sobre la Gestión de la Calidad.

Dados todos estos problemas y aprovechando la evolución de las TIC, el Proyecto Calidad de la Facultad 9 de la UCI ha decidido diseñar y construir una aplicación que permita gestionar algunos procesos imprescindibles en el logro de la calidad de los productos de software que se desarrollan, publicando la información necesaria para la ejecución de estas tareas y proponiendo una forma más rápida y cómoda de realizar las planillas que se generan.

Según lo expuesto anteriormente se identificó el siguiente **problema a resolver**: difícil acceso a la información, retrasos y escaso control detectado en la Gestión de los Recursos Humanos, y de las Auditorías y Revisiones en el grupo de Calidad de la Facultad 9.

Para lograr agilizar la gestión de la información en cada una de estas áreas. Se definió el **objeto de estudio**: los procesos para la Gestión de la información en el Grupo de Calidad de la Facultad 9, enmarcándose en el **campo de acción**: automatización de los procesos en los Módulos de Auditoría y Revisiones y Gestión de Recursos Humanos en un producto de software.

INTRODUCCIÓN

Se planteó como **objetivo general**: implementar los módulos de Auditoría y Revisiones y el módulo de Gestión de Recursos Humanos del grupo de Calidad de la Facultad 9.

Se elaboró la siguiente **idea a defender**: si se implementa un software que gestione los Módulos de Auditoría y Revisiones y Gestión de Recursos Humanos en la Facultad 9 permitirá desarrollar los procesos de estas actividades con mayor rapidez y propiciará un mejor acceso a la información.

Para resolver el problema y cumplir con el objetivo propuesto se identificaron varias **Tareas de investigación**:

1. Actualizar el conocimiento en torno al proceso de gestión de la información referente al módulo Recurso humanos.
2. Determinar el tipo de aplicación que se va a desarrollar.
3. Determinar las herramientas para el desarrollo de la aplicación.
4. Determinar el lenguaje de programación para el desarrollo de la aplicación.
5. Implementar el módulo Recursos Humanos.

Se propusieron como posibles resultados:

- Diagrama de Componentes.
- Código Fuente.
- Casos de Prueba.
- Prototipo Funcional de los módulos RRHH y AR.

Se utilizarán para el desarrollo de la investigación los siguientes métodos científicos:

Dentro de los métodos teóricos:

Histórico-Lógico.

Se utilizará este método para enunciar la Evolución Histórica de las aplicaciones web, los portales, portales de Calidad desarrollados ya sea a nivel mundial, nacional o en la Universidad, enfatizando en las aplicaciones de estos y resultados obtenidos.

Método Sintáctico-Analítico.

Este método facilitará el estudio de los procesos que se analizan puesto que permite dividir el fenómeno para así comprenderlo mejor y sacar de una manera más ágil y eficiente sus características.

También permitirá unir todo lo analizado previamente y establecer las relaciones y determinar las características generales necesarias para nuestro trabajo.

Entre los métodos empíricos:

Se utilizarán las encuestas.

Existen dos tipos de estas:

Directa: cuando la unidad estadística se observa a través de la investigación propuesta registrándose en el cuestionario.

Indirecta: cuando los datos obtenidos no corresponden al objetivo principal de la encuesta pretendiendo averiguar algo distinto o bien son deducidos de los resultados de anteriores investigaciones estadísticas.

Se hará uso de las encuestas directas para probar si la aplicación que se construya agiliza o no los procesos que se gestionan, estas serán aplicadas antes de comenzar a utilizar el sistema, para saber cómo era el proceso manual, y después de concluido el mismo para saber cómo es el proceso utilizando dicha aplicación y determinar si se cumple con la idea a defender propuesta. Se tomará una muestra de 3 profesores del Grupo Calidad de la Facultad 9 vinculados a las actividades de este proyecto.

Como aporte práctico, se les brindará a los usuarios del grupo de Calidad de la Facultad 9 y de otras facultades involucradas en los procesos de gestión de la información referente a la Calidad, una aplicación con las funcionalidades que le permitirán agilizar dichas actividades. Estará disponible además el acceso a toda la información necesaria para cada módulo y posibilitará además participar en un foro de discusión, donde podrán ventilar o aclarar dudas entre estudiantes y profesores experimentados en el tema o profesores vinculados a los Grupos de Calidad.

El presente documento cuenta con una estructura dividida en tres partes fundamentales, estas tres partes o etapas compondrán el desarrollo del trabajo, las cuales se mencionan a continuación:

Capitulo 1.Fundamentación Teórica: En este capítulo se aborda sobre el impacto que causan hoy en día las aplicaciones web, algunos conceptos asociados al dominio. Se mencionan las herramientas y lenguajes posibles a utilizar en el desarrollo del trabajo.

Capitulo 2. Construcción de la Aplicación: Aquí se muestra el estándar de codificación a utilizar, se analizará el diseño obtenido de la fase de Análisis y Diseño y se presentarán las funcionalidades más importantes con su implementación.

Capitulo 3. Verificación del Software: En este capítulo se verificará que el software cumpla con los requerimientos planteados, para ellos se diseñarán casos de prueba y posteriormente se harán las mismas que evaluaran el software.

Palabras claves: Calidad, Productos, Software, TIC, Proyecto, Métodos, Proceso, Gestión.

CAPÍTULO 1: Fundamentación Teórica.

1.0 Introducción

En este capítulo se justifica el tipo de aplicación escogida, teniendo en cuenta las necesidades del sistema que se quiere construir, así como el gestor de base de datos escogido. Se estudiarán los principales tipos de lenguajes y herramientas que se podrían usar para la realización de la aplicación, con el fin de realizar comparaciones y justificar la elección que se efectúe, también se abordará sobre las metodologías y/o procesos que puedan ser empleadas para garantizar y guiar dicha tarea.

1.1 Aplicaciones web.

Las aplicaciones web son del tipo cliente servidor ya que los datos y el procesamiento de la información se realiza solo en una computadora, llamada servidor y en otra computadora, la del usuario o cliente, la cual se comunica con el servidor enviando y recibiendo la información mediante un navegador web. Esto permite que un mismo servidor pueda atender las peticiones de gran cantidad de clientes al mismo tiempo, así todos estos podrán intercambiar información y utilizar la misma aplicación, teniendo como único requerimiento un navegador web.

Actualmente poder mostrar y acceder a la información de una empresa a través de Internet es una gran ventaja competitiva. Con una aplicación web podrá manejar inventarios, facturación, publicación de información con accesos restringidos a ciertos usuarios, mantenimiento y actualización de su sitio web, incorporación de elementos multimedia que lo harán más dinámico y divertido.

1.1.1 Ventajas

Las Aplicaciones Web facilitan el trabajo a distancia.

No se requieren combinaciones complicadas de Hardware/Software para utilizarlas. Solo un computador con un buen navegador Web, lo que permite la reducción en el costo de inversiones.

Son fáciles de usar. No se necesita de conocimientos avanzados en computación.

Brindan una alta disponibilidad ya que teniendo acceso a Internet puedes conectarte a ella en cualquier parte del mundo y a la hora que desees para consultarla.

Funcionan en cualquier Sistema Operativo. Su desarrollo es económico, sencillo y rápido.

Por tanto, la utilización de estas aplicaciones permite reducir costos y complicaciones, siendo una excelente opción para automatizar procesos, invirtiendo menos en equipo, desarrollo y capacitación.

1.1.2. Beneficios

Las aplicaciones web permiten reunir las diferentes áreas de la empresa.

Proporciona mayor control de datos y mayor seguridad en las distintas secciones del sitio web.

Permite un avanzado sistema de consultas, altas, bajas, y modificaciones de datos provenientes de cualquier área de la empresa, lo cual mantendrá la información actualizada en todo momento.

Otorga flexibilidad a la hora de determinar niveles de acceso según la confidencialidad de los datos.

A manera de resumen una aplicación web permite conectarse desde cualquier lugar y a cualquier hora, siempre y cuando se tenga red en la computadora y no será necesario actualizar cada vez que se efectúen cambios, ya que estos se llevan a cabo en el servidor. Posibilita que se pueda trabajar desde cualquier sistema operativo gracias a su característica multiplataforma, provee un mayor control de los datos al encontrarse estos en un solo computador (servidor). También permitirá publicar información del proyecto y temas relacionados con la calidad, por lo que estaña disponible para todo aquel que pueda necesitar de esta. Además dichas aplicaciones son sencillas de usar, aún para aquellos que no tengan conocimientos de computación, por lo que cualquier persona podrá trabajar en ella. Todos estos aspectos son necesarios en el sistema que se desarrollará en el Grupo de Calidad de la Facultad 9, de ahí que se escoja este tipo de aplicación.

1.2 Portal Web.

Un portal es un término, sinónimo de puente, el cual tiene la función de servir como un sitio principal de partida para aquellos que se conectan al *World Wide Web*. Estos tienden a ser visitados como sitios ancla por los usuarios. Los portales son muy reconocidos en internet por el poder de influencia que tienen sobre grandes comunidades. El propósito de estos sitios es localizar la información y los sitios que puedan interesar a los navegantes, para que a partir de este punto puedan comenzar su actividad en la red. Un portal es más bien una plataforma de despegue para la navegación en la web. Por tanto un sitio web no alcanza el rango de portal por el simple hecho de contener información relevante o por ser robusto.

Un portal es la portada de un grupo de sitios web individuales que comparten la temática de su contenido, los cuales resumen la información relevante de cada uno de estos, y que permite tener un panorama global de lo que está pasando. En los portales se pueden encontrar servicios de chat, correo electrónico, motores de búsqueda, información actualizada sobre los temas que trate, para facilitar al usuario la navegación y búsqueda de documentación e información en internet.

Los servicios y contenidos son dos aspectos importantes para lograr la fidelización del cliente en un portal. La idea de portal web ó portal de internet nace ante el problema de brindar a un grupo de usuarios el acceso a gran cantidad de servicios informáticos y recursos de forma integrada y sencilla. (3)

1.2.1. Clasificación de los portales web, atendiendo al usuario.

Portales Horizontales.

El objetivo de estos son los usuarios en general. Suelen ofrecer motores de búsqueda, correos y otras formas de comunicación. Las ganancias de dinero están sustentadas por los anuncios.

Portales Verticales.

Se especializan en determinados temas, buscan públicos objetivos muy determinados. Estos se pueden clasificar en función de sus objetivos:

- Portal Intranet: Comunicación corporativa para los empleados.
- Portal Extranet: Comunicación Corporativa para los proveedores / partners[1].
- Portal Vertical: Comunicación Corporativa con clientes.

Todo lo antes mencionado justifica la elección de un **portal**, específicamente **intranet** que brinde a los usuarios acceso a una gran cantidad de servicios informáticos y recursos de forma integrada y sencilla, presentando además información actualizada del tema que trata. De esta forma se podrá publicar información para toda la comunidad universitaria UCI, información que pueda serle útil a todo el personal relacionado con los temas que se publiquen, así como a aquellos proyectos involucrados. Además un portal permite la suscripción de usuarios lo que posibilitará mantener contacto con los usuarios que se registren y que estos den criterios sobre lo que les gusta del sitio, o recomienden lo que se puede agregar o cambiar. Otros aspectos importantes son que proporcionarán mayor organización y facilitarán a los usuarios la navegación y la gestión de la información, ya que no necesitarán de conocimiento técnicos para esto. Un portal que sirva como puerta de enlace a otros sitios importantes para el navegante.

[1] s. socio, asociado, compañero, partenaire; colega, cónyuge; participe, coadjutor, coadtutor, fautor

v. desempeñarse en calidad de socio

1.3 Herramientas de Edición Web.

1.3.1 Zend Studio.

Es un programa de la casa Zend, quienes son impulsores de la tecnología de servidor PHP. Este proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código, además de servir como editor de texto. Permite agilizar el desarrollo web y simplificar proyectos complejos. Fue escrito completamente en JAVA (Just Another Vague Acronym), lo que supone que en ocasiones no funcione con buena velocidad como otras aplicaciones de uso diario. No obstante esto le ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para sistemas operativos (SO) como Windows, Linux, MacOS, aunque el desarrollo de las versiones de este último sea un poco más lento. Zend Studio consta de dos partes, estas son funcionalidades en la parte del cliente y del servidor. Cada parte se instala por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

El Editor

La parte que permite escribir los scripts es bastante útil para la programación en PHP. Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Ofrece también las ayudas típicas de los editores avanzados a la hora de escribir, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a que elementos corresponden los inicios, y cierres de las etiquetas, paréntesis o llaves, moverse al principio o final de una función, indentación automática del código, etc.

Gestión de proyectos

La barra de la izquierda, que permite navegar los archivos de nuestro ordenador, también dispone de herramientas para gestionar los proyectos, muy útiles para mejorar la productividad en la programación. Los proyectos permiten guardar mucha más información al programa sobre los archivos, discos, servidores, etc., que se gestionen en nuestras aplicaciones PHP.

Después que los archivos se han añadido al proyecto se pueden guardar señales como punto de ruptura en las depuraciones, asimismo al poner Zend Studio en marcha, se abren los archivos que se utilizaban al cerrar y las herramientas de completar código mejoran sus comportamientos, asumiendo toda la información de los archivos relacionados con el proyecto.

Zend Studio implementa además interesantes opciones de trabajo en grupo, al integrar el sistema de trabajo conocido como CVS (Soporte para control de versiones).

La herramienta de depuración.

Zend Studio dispone de una herramienta muy interesante de debug o depuración. Gracias a esta se puede ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión. Se pueden colocar puntos de paradas de los script y realizar las opciones típicas de depuración.

Además de la ventana que visualiza el contenido de las variables, tiene otras donde muestra la salida del script según se va generando y otra con las alertas y errores. Las posibilidades se completan con distintos tipos de depuración, en local, en remoto, o a partir de una URL.

1.3.1.1 Características, Ventajas y Desventajas.

Características: Brinda excelente completamiento de código, coloreado en la sintaxis del código, permite administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para Web Services, PHP4, PHP5 y SQL.

Ventajas: Agiliza nuestro trabajo, cuenta con un buen Depurador, infinitas opciones que permiten un desarrollo profesional de nuestras aplicaciones.

Desventajas: Requiere Licencia de pago, no incluye editor visual HTML.

1.3.2 Aptana Studio.

Apatana es un IDE para la programación de aplicaciones web dinámicas, con soporte especial para Ajax (*Acrónimo de Asynchronous JavaScript + XML*)/JavaScript, es uno de los mejores programas que utiliza Ajax.

Es un entorno de desarrollo basado en Eclipse que soporta HTML, (Document Object Model) DOM, JavaScript, CSS, así como soporte para plataformas emergentes que utilizan Ajax como Adobe AIR e iPhone.

Para los que han usado Eclipse, les será fácil acostumbrarse a su uso, y para los que no es bastante sencillo, además de que es completo.

Aptana Studio puede extender sus funcionalidades a base de Plug-ins, para trabajar más cómodo en algunos lenguajes de programación como PHP, Python o Ruby on Rails.
Aptana es totalmente gratuito y funciona en plataformas Windows de 32 o 64 bits, Mac OSX 10.4 o superior, en Linux de 32 bits con GTK o en Linux de 64 bits como plug-ins dentro de Eclipse.

Esta herramienta incluye XHTML, JavaScript, CSS, PHP, además de correctores para estos lenguajes, ilustra diferentes idiomas y funciones con colores diferentes para que puedan ser reconocidas fácilmente, también incluye dos ventanas para observar los cambios realizados en tu página en Firefox y en Internet Explorer.
Permite probar con otros navegadores para que los usuarios puedan ver cómo reacciona en los diferentes ambientes.
Otra ventaja que posee es que en él se puede debuggear, para analizar los resultados que van obteniendo las variables y las funciones, así como tiene también completamiento de código para ahorrar tiempo al programador y facilitarle su trabajo.

1.3.2.1 Características, Ventajas y Desventajas.

Características: Protocolos de comunicación FTP, editor JSON (Javascript Object Notation), protocolos de Comunicación FTPS, SFTP, soporte en Fórum.
Ventajas: Permite comprobar la compatibilidad de las funciones con los diferentes navegadores, multiplataforma, sincronización con carpetas locales y remotas, incluye plugins para Eclipse.
Desventajas: Consumo de recursos.

Justificación de la herramienta seleccionada.
Se decide trabajar con Zend Studio, por las facilidades que proporciona y todas la ventajas que posee, las cuales bien aprovechadas colaborarán con un buen producto. Esta herramienta permite debuggear mientras se esté programando, es decir se podrá ejecutar el código por partes para observar el funcionamiento, lo que facilitará identificar los errores que se presenten en el código. También destaca su completamiento de código lo que ahorra tiempo e incluso proporciona aquellas funciones que se olviden o no se conozcan. Otros aspectos importantes son la ayuda contextual que posee su editor de texto y las librerías de funciones, que ofrecen los nombres de las funciones y los diferentes parámetros que debe recibir. Destaca también el resaltado de sintaxis. Zend aumentará la productividad de desarrollo en PHP.

1.4 Lenguajes de programación Web y Tecnologías.

1.4.1 PHP

Acrónimo de *Personal Home Page*. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de información y mucha documentación. PHP se escribe dentro del código HTML, haciéndolo fácil de utilizar, al igual que el ASP *(Active Server Pages)* de Microsoft pero con las ventajas de ser rápido, gratuito, multiplataforma y seguro. PHP posibilita configurar los servidores de forma tal que se permitan o rechacen diferentes usos, lo que puede hacer el lenguaje seguro dependiendo de las necesidades de cada quién. PHP fue creado en 1994 por Rasmus Lerdorf, pero como está desarrollado en política de código abierto, ha recibido gran cantidad de contribuciones en sus años de existencia.

Una capacidad importante de PHP es la compatibilidad con las bases de datos más comunes, como MySQL, mSQL, ORACLE, Informix y ODBC (Open Database Connectivity), por citar algunos ejemplos. Incluye funciones para envío de correo electrónico, carga de archivos, creación de imágenes dinámicamente en el servidor con formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

Este lenguaje está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones que este posee. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de redes.

PHP ejecuta más rápido las operaciones matemáticas que ASP.

1.4.2 JSP

JSP es el acrónimo de *Java Server Pages*, es una tecnología orientada a crear páginas web con programación en Java. Lo que es una ventaja ya que esta ayuda en el manejo de memoria protegiéndola contra fallos y el duro trabajo de buscar los fallos de pérdida de punteros de memoria que pueden hacer más lento el funcionamiento de una aplicación.

Utilizando JSP se puede crear aplicaciones web que se ejecuten en varios servidores de múltiples plataformas, debido a que Java es en realidad un lenguaje multiplataforma. JSP se puede ejecutar en los sistemas operativos y servidores web más populares, así como Netscape, Apache o Microsoft IIS. Las páginas JSP están compuestas de código HTML/XML mesclado con etiquetas especiales para programar

script de servidor en sintaxis Java, por lo que estas páginas se pueden escribir con los editores de código de HTML/XML habituales.

El motor de las páginas JSP está basado en los servlets[2] de Java.

En JSP las páginas se crean de forma semejante a como se hace en ASP y PHP.

El API JSP se beneficia de la comunidad Java.

Utiliza la combinación de tags y script para crear páginas web dinámicas. Además permite a los desarrolladores crear tags propios para no depender tanto de los scripts.

Los componentes JSP son reusables en distintas plataformas (Unix, Windows).

En este ejemplo se ve una página muy simple de JSP, y como esta se convierte en un servlet.

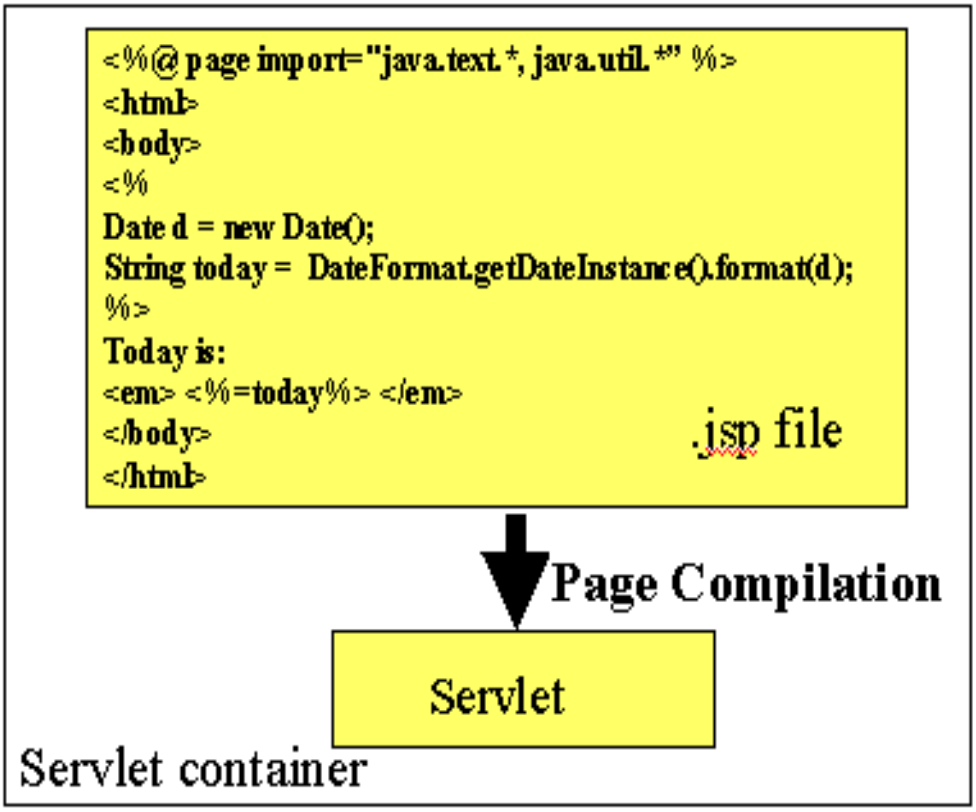

Figura # 1 Página simple de ASP.

[2] Programas en Java destinados a ejecutarse en el servidor.

1.4.3 ASP

Es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. Las *Active Server Pages* son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, script y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web.

Su implantación está limitada para arquitecturas basadas en tecnología Microsoft. Solo tiene soporte nativo para IIS y Personal Web Server que son los dos servidores web para servidores Microsoft, el primero con tecnología NT(New Technology) y el otro para sistemas Windows 98 y similares. Esta tecnología es específica de Microsoft que desarrolla sus procesos internamente. Usa la combinación de tags y scripts para crear páginas web dinámicas al igual que JSP. Usa VBScript (Visual Basic Script Edition) o Jscript como lenguaje de script. Los ASP sobre NT son susceptibles a sufrir caídas. El mantenimiento de este lenguaje no es de lo más rápidos.

1.4.4 Java Script.

Lenguaje de programación que se encuentra en las páginas web, con el se define el comportamiento de estas páginas, de forma que enriquece la experiencia del usuario al ofrecerle una interacción con el directamente desde el navegador, lo que da una velocidad de respuesta muy alta.

Es un lenguaje interpretado, no requiere compilación, por lo que se puede ver su código y por tanto su funcionalidad desde el lugar donde se ejecuta, siendo muy fácil capturar y readaptar para nuestras necesidades, de aquí su rápida evolución en internet. Este lenguaje se ha hecho imprescindible para todo desarrollador / diseñador web. Actualmente es muy usado para las aplicaciones web. Java Script es usado además dentro de la tecnología AJAX. Es soportado por la mayoría de los navegadores, como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. Sus sintaxis son similares a las de Java y las de C, no necesita de ningún framework ya que al ser un lenguaje del lado del cliente es interpretado por el navegador.

Permite la programación orientada a objetos, la declaración de variables, arreglos, la utilización de ciclos y de for, el trabajo con funciones propias del lenguaje y otras que sean declaradas por los usuarios, así como el uso de comentarios de una y de varias líneas.

Ventajas

Este lenguaje es incorporado en cualquier página web, se puede ejecutar sin depender de otro programa para su visualización.

1.4.5 AJAX

Su concepto es: cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son utilizados para actualizar la página solo renderizando la misma y mostrando u ocultando porciones de esta.

AJAX no es una tecnología, sino muchas tecnologías unidas. Incorpora:

- Presentación basada en estándares usando XHTML (Hipertext Markup Languaje Extensive) y CSS.
- Exhibición e interacción dinámica usando el Document Object Model.
- Intercambios y manipulación de datos utilizando XML (Extensive Markup Languaje) and XSLT (Extensible Stylesheet Languaje Transfomations).
- Recuperación de datos asincrónica utilizando XMLHttpRequest.
- Y JavaScript poniendo todo junto.

Una aplicación AJAX elimina la naturaleza “arrancar-frenar-arrancar-frenar”, de la interacción en la web introduciendo un intermediario en la red -un motor AJAX- entre el usuario y el servidor.

Al iniciar sesión, en lugar de cargar una página, el navegador carga el motor AJAX, escrito en Javascript y usualmente “sacado” en un frame oculto. Este motor se hace responsable de renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario. Este motor permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). De esta forma el usuario nunca estará mirando una página en blanco del navegador y un icono de reloj de arena esperando a que el servidor responda.

Algunos productos realizados por Google son aplicaciones **AJAX**, son estos: Gmail, Google Maps, Google Groups, Orku y Google Suggest.

Desventajas

El uso de AJAX puede producir lentitud en el navegador, ya que mientras más código Java se utilice, más código JavaScript será necesario del lago del navegador, lo que disminuye el rendimiento del equipo.

1.4.6 HTML

HTML es el formato primario utilizado en la web mundial, a pesar de no ser un verdadero lenguaje, es sencillo. Exhibe las páginas con gran variedad de colores, de formas y de objetos. Estas siglas corresponden a “Lenguaje para marcado de hipertexto”, para mejor entendimiento, se trata de un lenguaje para estructurar documentos a partir de texto en World Wide Web. Dicho lenguaje se basa en tags (instrucciones que definen como se mostrará el texto) y atributos (parámetros que dan valor al tag).
La mayoría de estos documentos poseen estructuras comunes (títulos, párrafos, listas, tablas), dichas estructuras van a ser definidas por este lenguaje mediante los tags, todo lo que no sea tag o se encuentre dentro de un tag, es parte del documento mismo.

HTML no describe la apariencia del diseño de un documento, sino que permite a cada plataforma establecer un formato según su capacidad y la de su navegador. Es por esto que no se debe diseñar los documentos en función de cómo se mostrarán en el navegador, sino que hay que centrarse en brindar un contenido claro y bien estructurado, que resulte fácil de leer y entender, de esta forma no habrá frustraciones.
Ventajas del lenguaje: Compatibilidad. Facilidad de aprendizaje debido al reducido número de tag que usa.

Justificación del lenguaje seleccionado.
Se opta por utilizar PHP como lenguaje de programación web para el desarrollo de la aplicación, ya que este es rápido, gratuito, multiplataforma, seguro y demás es fácil de aprender. También se tuvo en cuenta que se poseen conocimientos sobre este lenguaje. Se aprovecha además las posibilidades de configuración que ofrece y la compatibilidad con diferentes bases de datos, ya que contiene funciones para el desarrollo y sencillo trabajo con estas, así como las librerías de funciones que este brinda, las librerías de información y la gran documentación que incorpora. Es soportado por el servidor Apache, generalmente es usado como módulo de este. Está basado en el motor de interpretación de Zend. Utiliza sintaxis similares a las de C++ y Java, lenguajes de los cuales ya se tiene dominio. Está orientado a la web y permite utilizar técnicas de programación orientada a objeto.

1.5 Servidores Web.

Un servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan, y este las satisface. En dependencia del tipo de petición el servidor web buscará una página web o bien ejecutará un programa en el servidor. No obstante siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

Estos son fundamentales en el desarrollo de las aplicaciones del lado del servidor, server side applications, que se vaya a construir, ya que se ejecutarán en el mismo. (4)

1.5.1 Apache Server.

Apache es un servidor web de tecnología código abierto (open source), sólido y para uso comercial desarrollado por la Apache Software Foundation. Es flexible, rápido y eficiente, continuamente actualizado y adaptado a los diferentes protocolos. Es uno de los mayores logros del software libre.

1.5.1.1 Cara

cterísticas destacables.

- ✓ Multiplataforma.
- ✓ Es un servidor web respecto al protocolo HTTP/1.1.
- ✓ Modular: Se puede adaptar a diferentes entornos y necesidades, con los diferentes módulos de apoyo que brinda, y con la API de programación de módulos, para desarrollar módulos específicos.
- ✓ Basado en hebras en la versión 2.0.
- ✓ Incentiva la realimentación de los usuarios, donde se obtienen nuevas ideas, informes de fallos, y parches para la solución de estos.
- ✓ Se desarrolla de forma abierta.
- ✓ Extensible: gracias a ser modular, se ha podido desarrollar diversas extensiones, entre las que destaca PHP. (5)

1.5.2 Xitami

Xitami es un potente servidor web, gratuito, código abierto, fácil de instalar y configurar, destinado además para ser seguro, por lo que puede negar el acceso a un determinado ip. Permite crear una cuenta de ftp para cada usuario con sus respectivos derechos de acceso, y se puede manejar todos los datos de forma virtual.
Servidor de alta calidad, portátil y libre.

Este servidor consume pocos recursos y presenta una interfaz altamente intuitiva. Posibilita la gestión de usuarios y contraseña *on-the-fly* (los cambios se hacen sin tener que reiniciar el servidor).

1.5.2.1 Características

Soporte para el HTTP (*Hypertext Transfer Protocol*)/1.0, FTP (*File Transfer Protocol*), CGI (*Common Gateway Interface*)/1.1, SSI (*Server Side Includes*).
Proporciona establecimiento de alta seguridad de registros: CLF (Formato Común de Registro), IIS (*Internet Information Server*) y XML (*Extensive Markup Languaje*).

1.5.3 Internet Information Server (IIS).

IIS es un potente servidor que permite la gestión en profundidad de los sitios web, ya que facilita instrumentos de desarrollo para potenciar sus posibilidades.
Entre sus características destaca la presencia del protocolo HTTP 1.1 que ofrece sensibles mejoras de las prestaciones, disminuyendo los tiempos de respuesta.
Su versión 6.0 ofrece una estructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web sobre todas las versiones de Windows Server 2003. Este servidor hace posible que las organizaciones aumenten la disponibilidad de sus aplicaciones y sitios Web y a la vez reducir sus costos administrativos. IIS 6.0 soporta la Iniciativa de Sistemas Dinámicos (DSI)[3], con monitorización de estado de salud automático, aislamiento de procesos y capacidades de gestión mejoradas.

Justificación del servidor seleccionado.

[3] La Iniciativa de Sistemas Dinámicos (DSI), es un compromiso de Microsoft y sus partners para crear sistemas dinámicos capaces de auto-administrarse.

Por todo lo descrito anteriormente, y la gran cantidad de características destacables que presenta, se define Apache como el servidor para montar la aplicación que se desarrollará. Se elige este para trabajar, fundamentalmente por la integración que tiene con el lenguaje PHP, lenguaje escogido para la implementación del portal. Teniendo también en cuenta su flexibilidad, rapidez y eficiencia, que es de tecnología Código Abierto, sólido, actualizado continuamente y adaptado a los diferentes protocolos. Es además multiplataforma, fácil de adaptar a los diferentes entornos y necesidades, incentiva la realimentación del usuario, extensible, y se desarrolla de forma abierta, ofrece una alta seguridad. Otro aspecto importante es que es el servidor más usado en Internet.

1.6 Base de datos.

Una Base de Datos (BD) es un conjunto de información, la cual se encuentra organizada de forma que un programa de computadora pueda obtener los fragmentos de datos que le sean necesarios. Una base de datos es un archivo de datos electrónico. Una base de datos habitual se organiza por campos, registros y archivos. Un campo es una pieza única de información, un registro es un sistema completo de datos, y un archivo es una serie de registros.

1.6.1 PostgreSQL

Es un avanzado sistema de gestión de base de datos objeto-relacional (ORDBMS), ya que incluye características orientadas a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, aunque PostgreSQL no es un sistema de base de datos puramente orientado a objetos. Está basado en el proyecto Postgres, está bajo licencia BSD (Berkeley Software Distribution).

Es una derivación libre (código abierto) de este proyecto, utiliza lenguaje SQL92/SQL99. PostgreSQL puede manejar múltiples conexiones concurrentes de los clientes

PostgreSQL fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual.

Principales Características.

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además de los base, también puede soportar datos de tipo fecha, monetario, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. Permite la creación de tipos propios.

- Incorpora una estructura de datos array (arreglos).
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite declarar funciones propias, así como definir disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen) por lo que a este gestor de base de datos se le incluye entre los gestores objetos relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos

Se procede a trabajar con Postgres para la realización y montaje de la base de datos, primeramente porque es software libre, y por las ventajas que este posee, como permitir la herencia entre tablas, la cual es una característica de la orientación a objetos, por lo que se incluye entre los gestores objetos relacionales. Incorpora el trabajo con arreglos y diversas funciones, además de que permite crear otras funciones propias, también soporta gran cantidad de tipos de datos, destacando entre ellos los de redes y de fechas, y posibilita la gestión de diferentes usuarios, así como la asignación de sus permisos.

1.7 Framework

El concepto de Framework es muy genérico, se refiere a ambiente de trabajo, y ejecución. Los framework en general son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambientes de ejecución).

Los Frameworks orientados a objeto (o simplemente frameworks) son considerados la piedra angular de la ingeniería de software moderna. Estos han ganado aceptación rápidamente debido a su capacidad para promover la reutilización de código del diseño y el código fuente.

Son generadores de aplicaciones que se relacionan con un dominio específico directamente, o sea con una familia de problemas que se relacionan entre sí.

1.7.1 Características de los Framework.

Reutilización de diseño y de código.

Cuando se selecciona un framework se reutiliza tanto el artefacto físico (código y ejecutable) como el diseño que este trae incluido.

Principio de inversión del control.

El código de aplicaciones usuales está formado por piezas principales que utilizan clases o rutinas (prácticas, repetición) utilitarias (económica, ventajosa) que son externas.

Sin embargo los módulos o clases de las aplicaciones basadas en Framework son manejados (manipulados) por la técnica de control interno de estos.

1.7.2 Ventajas de la utilización de un framework.

Modularidad y reducción de la complejidad.

La aplicación está integrada por subsistemas especializados en diferentes aspectos esenciales de toda aplicación (presentación, persistencia, manejo de log).

Fortaleza al cambio.

Los módulos se pueden cambiar y evolucionar sin afectar la arquitectura global de la aplicación.

Documentación

Su documentación promueve el uso correcto de este y minimiza el esfuerzo necesario para el mantenimiento.

Estructura

Cuando se trabaja sobre un framework, se adopta una estructura donde a la hora de construir la aplicación, el desarrollador no tiene que decidir el cien por ciento del diseño de la aplicación.

Distribución de funciones.

Permite paralelizar el trabajo de desarrollo ya que la solución puede realizarse como un conjunto de piezas que encajarán en el framework usado.

Eficiencia

El desarrollador se puede concentrar en los requerimientos funcionales de la aplicación.

Aplicaciones ricas.

Posibilidad de brindar más funcionalidades a los usuarios de estas.

En el ambiente empresarial.

- La reutilización permite diseñar e implementar a costos más bajos y con mayor calidad.
- Brinda el dinamismo que requiere el negocio e incrementa el nivel de calidad en el proceso de desarrollo.
- La reutilización en las compañías es imprescindible para realizar procesos ágiles.

1.7.3 Principales Framework.

1.7.3.1 Symfony

Symfony es el mejor framework para crear aplicaciones PHP y la forma más sencilla de aumentar la productividad y calidad de tu trabajo. Symfony ha sido probado con éxito en algunos de los sitios web más grandes del mundo. (6)

Symfony es un Framework completo, el mejor, el más sofisticado y documentado. Un framework de por si brinda productividad, calidad en la programación, mantenimiento, a lo que Symfony agrega rendimiento de la aplicación.

Symfony está dividido en 4 entornos:

- ✓ Desarrollo(dev)
- ✓ Pruebas
- ✓ Intermedio
- ✓ Producción(prod)

Gracias a sus características, Symfony optimiza el desarrollo de aplicaciones web. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Provee varias herramientas y clases orientadas a disminuir el tiempo de desarrollo de una aplicación web. Automatiza las tareas más comunes, lo que permite que al desarrollador que se pueda dedicar por entero a los temas específicos de cada aplicación. Todas estas ventajas traen como resultado que no se tenga que reinventar la rueda cada vez que se construye una aplicación nueva.

Symfony está desarrollado completamente con PHP5, es compatible con la mayoría de gestores de bases de datos y es multiplataforma.

Características

Fácil de instalar y configurar. Es independiente del sistema gestor de base de datos, sencillo de usar. Posee flexibilidad para adaptarse en los casos más complejos. Está basado en la premisa de “convenir en vez de configurar”, donde el desarrollador solo deberá configurar aquello que no es usual. Cumple con gran parte de las mejores prácticas y patrones de diseño para web. Está preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de casa empresa. Propicia fácil lectura del código, permitiendo un mantenimiento sencillo. Fácil de extender, esto permite su integración con librerías desarrolladas por terceros.

1.7.3.2 Zend Framework.

Este es otro de los framework que existen para el desarrollo de aplicaciones web en PHP. Está desarrollado en política de código abierto, se genera código orientado a objeto. Este framework ofrece un gran rendimiento y proporciona una robusta implementación del patrón MVC.

Ventajas

Reduce el tiempo de desarrollo de las aplicaciones, estandariza los procesos más frecuentes, dotándolos de gran robustez, facilita el mantenimiento de las aplicaciones.

Ofrece muchas facilidades para el acceso a recursos avanzados que de otro modo resultan bastante más costosos de desarrollar.

A diferencia de otros, es posible utilizarlo en modo "desacoplado", es decir, aquellas clases o componentes que sean necesarios en cada proyecto, sin arrastrar todo el framework detrás para cualquier pequeña necesidad.

Tiene el respaldo de la propia ZEND, creadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP.

1.7.3.3 Prado

Prado es un framework de desarrollo rápido de aplicaciones web en PHP 5, además orientado a objeto, lanzado bajo licencia revised BSD, gratuito. Entre los objetos que esta plataforma de desarrollo ofrece, se encuentra: acceso a bases de datos, formularios y controles web entre otros. Además Prado soporta componentes AJAX, permite personalización en la localización de errores y excepciones e incorpora seguridad contra XSS.

Prado es fácil de usar, permite la reutilización, propicia el buen funcionamiento y permite también la separación del contenido y la presentación.

Tabla 1 Comparación entre los frameworks más usados y actuales.

Frameworks	Características	Sistema Operativo	Ventajas
Symfony	Código abierto. Completo, el más sofisticado. Sencillo de usar, de fácil instalación y configuración.	Multiplataforma	Paradigma de la reutilización de código. Compatible con la mayoría de gestores de base de datos. Gran cantidad de documentación.
Prado	Gratuito, también puede ser usado para crear aplicaciones comerciales. De desarrollo rápido, orientado a objetos.	Multiplataforma	Reutilización, fácil uso, funcionamiento, integración.
Zend Framework	Código abierto. Orientado a objetos.	Multiplataforma	Reduce tiempo de desarrollo, propicia gran robustez, facilita el mantenimiento de aplicaciones, permite utilizar sus componentes por separado.

Justificación del Framework Symfony.

El portal se desarrollará sobre el Framework Symfony, primeramente aprovechando que tiene disponible gran cantidad de documentación, es fácil de aprender, instalar y configurar, además por la característica que poseen los Framework de ser software libre, así como por proporcionar la reutilización de componentes, lo que posibilita una programación rápida del sitio, con una interfaz aceptable y porque propone una alta seguridad. Permite además la optimización en el desarrollo de las aplicaciones, la integración con la base de datos que se seleccione, brindando también mantenimiento y uso sencillos, siendo flexible en los casos más complejos. También propicia la automatización de las tareas más comunes, lo que permitirá dedicarse por entero a los temas específicos de la aplicación.

1.8 Antecedentes de la Investigación.

Antes de comenzar la implementación del portal se analizaron algunos de los sitios o portales existentes en la universidad. Se tuvo en cuenta que guardaran relación con las labores de obtención de la calidad, principalmente con la gestión de los módulos de recursos humanos y auditorías y revisiones. De estos sitios se detectó que realizan o desempeñan diferentes funciones a las que se desean implementar en el grupo de calidad de la facultad 9, por lo cual se decidió construir un portal que gestione la información resultante de las acciones ejecutadas como parte de los módulos de RRHH y AR.

1.9 Tendencias Actuales.

Las tendencias actuales muestran a Internet (la gran red de redes) como la principal vía para poner los servicios de cualquier empresa u organización a disposición del público o clientes de todo el mundo, así como para sus propios trabajadores, en internet las aplicaciones web tienen un papel protagónico para lograr esto. Otra tendencia que se puede percibir hoy es el uso del **Software Libre**, lo que ha tomado gran auge a nivel mundial, por las ventajas que este brinda. Cuba se ha integrado al movimiento del software libre y se hace todo lo posible por generalizar su uso, teniendo en cuenta que se camina hacia el socialismo, por lo que optar por esto sería lo más lógico. La UCI apoya y desarrolla esta idea, por lo que tiene creada una comunidad de software libre, donde se puede conocer más sobre esta alternativa, y tiene además una facultad que trabaja con dicho tipo de software. De ahí que se haya hecho énfasis en este concepto a la hora de seleccionar cada aspecto.

1.10 Conclusiones Parciales.

Se identificaron los diferentes y principales aspectos que se utilizarán para la creación del portal, abarcándose las definiciones, características, ventajas, y en algunos casos los beneficios que estos traen al ser utilizados. Se espera que haya quedado clara cada justificación realizada, de no ser así se puede consultar las tesis de los restantes roles implicados en la realización de este portal.

Después de concluir este capítulo se procede a la creación de una aplicación web, específicamente un portal vertical (intranet), en el cual se pueda brindar servicios a la comunidad universitaria, así como otras organizaciones o instituciones que necesiten del mismo.

CAPÍTULO 2: Construcción de la Aplicación.

2.1 Introducción

En este capítulo se aborda primeramente sobre el estándar de codificación definido para la implementación, el diagrama de componentes creado para representar la relación entre los componentes existentes. Se analiza exhaustivamente el diseño realizado por el analista y se tienen en cuenta los cambios necesarios de acuerdo a la implementación que se quiere realizar. Se describen los algoritmos más importantes de la aplicación.

2.2 Estándares de codificación.

Un estándar de codificación comprende los aspectos de la generación de código y repercute directamente en la legibilidad y la extensibilidad de cualquier proyecto de software, haciendo que nuevos desarrolladores se acoplen rápidamente al proceso de desarrollo. Usar estándares de codificación sólidos y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento del mismo.

Para la selección de los estándares a utilizar en la programación del portal web de gestión de información se consultaron tres documentos. No se escogió ninguno de los analizados porque no se ajustan a las necesidades que se tienen, por tal motivo se creó un estándar propio para la programación del portal web de gestión de información.

El estándar definido a continuación tiene como objetivo lograr un diseño de programación homogéneo en el transcurso de la aplicación, que permita un fácil y rápido entendimiento por parte de los involucrados.

2.2.1 Indentación

La unidad de indentación de bloques de sentencias es de 4 espacios.

Sentencias de código

Cuando las sentencias sean muy largas se deben fraccionar atendiendo a los siguientes principios generales:

Fraccionar después de una coma “,”.

Fraccionar después de un operador “+”,”-“,”%”.

Es aconsejable romper unidades de alto nivel y no las de bajo nivel, ejemplo:

```
Cant_max = cant_1 + (cant_2 + cant_3 – cant_4)
/ cant_5 * cant_6; //Aconsejable
Cant_max = cant_1 + (cant_2 + cant_3
– cant_4) / cant_5 * cant_6; // No aconsejable
```

2.2.2 Comentarios

Los comentarios deben añadir claridad al código.

-Existen varios tipos de comentarios, los simples y los compuestos.

Simples:

Una sola línea de código.

Ejemplo: // esta clase devuelve la posición del objeto que se pasó/

Compuestos:

Tienen más de una línea de código.

Ejemplo:

```
/* public void agregar (persona p)
    {
    Bloque de instrucciones.
    }*/
```

2.2.3 Declaraciones

Las variables se declaran cada una en una línea distinta.

Ejemplos:

```
String var = “”; // Bien
String var_1 = “”; // Bien
String var = “”; String var_1 = “”; // Mal
```

2.2.4 Espacios en blanco.

2.2.4.1 Uso de líneas en blanco.

Se debe usar una línea en blanco:

Entre métodos.

Entre las variables locales de un método y la primera sentencia.

Antes de un bloque o un comentario de línea (para saber de una ojeada de que trata).

2.2.4.2 Uso de caracteres de espacio.

Estos deben usarse en las siguientes circunstancias:

- ✓ Después de una coma en lista de parámetros de un método.
- ✓ Entre operadores binarios +, -, *,=, etc. Nunca entre un operador uno-ario y su operando.
- ✓ La sentencia for y su paréntesis, ejemplo: for (exp1; exp2; exp3);

2.2.5 Asignación de nombres.

- ✓ Se usa un descriptor que deje clara la función de la variable, método o clase.
- ✓ Se usa terminología aplicable al dominio.
- ✓ Si se utilizan abreviaturas, debe quedar plasmado en algún lugar lo que significan.
- ✓ Evitar los nombres largos, menos de quince letras.
- ✓ Evitar nombres parecidos, que difieran en una letra o en el uso de mayúsculas.
- ✓ Los nombres no deben constar de más de dos palabras.
- ✓ No usar siglas en los nombres, a no ser que sean muy largos o que sean siglas claras, que todos conozcan.

2.2.6 Uso de las llaves.

Se debe utilizar una línea para cada llave.

Ejemplo:

```
public bool Buscarpersona (persona p)
{
    for (exp1; exp2; exp3)
        {
```

```
        if (exp)
                {
                Código;
                }
        }
        return var;
}
```

2.3 Estudio del diseño.

2.3.1 Análisis crítico y detallado del diseño entregado.

Luego de obtener los resultados de la fase de Análisis y Diseño, se puede proceder con la fase de Implementación, para ello se estudia primeramente el diseño elaborado por el analista. Luego se examina críticamente cada una de las funcionalidades necesarias y clases persistentes relacionadas con estas, garantizando la efectividad en el registro de información y comprobando que estén bien descritas y claras, lo que facilitará el trabajo a la hora de implementarlas. Se verifica que lo expuesto en estas descripciones de las funcionalidades sea posible de implementar. Se discuten las inconsistencias encontradas con el objetivo de realizar los cambios necesarios para obtener un modelo de datos acorde a lo pensado. Finalmente se determina que el diseño es el indicado y se puede continuar con las demás actividades de la fase de implementación.

2.3.2 Resultado del análisis realizado.

Se obtuvieron 16 casos de uso, de ellos 6 son críticos, siendo estos los más importantes y complejos a la hora de implementar, complejidad que se debe a que hay que realizar todas las operaciones con los datos, no por la dificultad del código. Se identificaron un total de 35 funcionalidades relevantes y 16 adicionales o de apoyo, para lograr el correcto y completo funcionamiento de la aplicación. Se realizaron ajustes en los datos representados en cada funcionalidad, obviando aquellos menos relevantes, para facilitar la introducción o almacenamiento de los mismos en el modelo. Fueron detectadas además algunas inconsistencias que podrían afectar el buen registro de información, por lo que de un diseño de base de datos de 45 tablas, se concluye en trabajar con 43 de ellas, ya que se acordó quitar algunas y

agregar otras, también se adicionaron y se eliminaron relaciones para facilitar las operaciones con la base de datos que se montará en el gestor PostgreSQL.

2.4 Implementación

Este nivel en la construcción de la aplicación debe ser consistente con el diseño anteriormente analizado, por lo que se construye la aplicación en términos de componentes, o sea: código fuente, ejecutables, scripts, entre otros. El resultado de esta etapa es el modelo de implementación, el cual relaciona componentes y subsistemas, para representar estos se realizan los diagramas de componentes.

2.4.1 Diagramas de Componentes.

Un diagrama de componentes es la implementación lógica de la implementación de un sistema, está constituido por los componentes y las dependencias entre ellos.

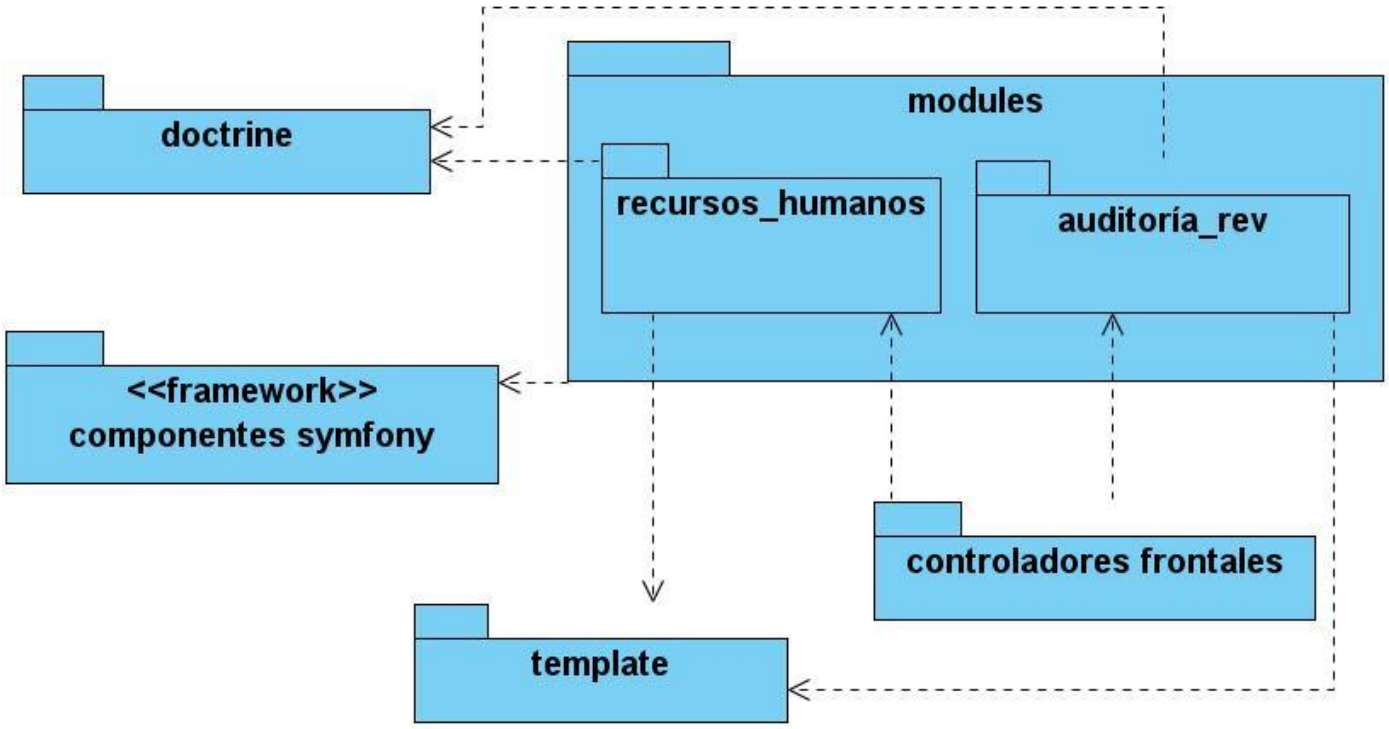

Figura # 2 Diagrama de componentes.

2.5 Estructura y organización de la aplicación.

Este framework proporciona organización en la aplicación que se desarrolla, ya que organiza el código fuente en una estructura de tipo proyecto y almacena los archivos del proyecto en una estructura estandarizada de tipo árbol.

Descripción de las carpetas que contiene el mismo:

apps: contiene un directorio por cada aplicación del proyecto.

cache: aquí se almacena la versión cacheada de la configuración.

config: guarda los archivos de configuración general del proyecto.

data: contiene todos los archivos relacionados con los datos.

lib: almacena las clases y librerías externas, normalmente código común de todas las aplicaciones del proyecto.

log: contiene los archivos log generados por Symfony, también los generados por el servidor web, bases de datos y otros componentes del proyecto.

pluggins: guarda los pluggins instalados en la aplicación.

test: guarda las pruebas unitarias y funcionales escritas en php, y compatibles con el framework de prueba de Symfony.

web: esta es la raíz del servidor web. Estos archivos son los únicos a los que se puede acceder desde internet.

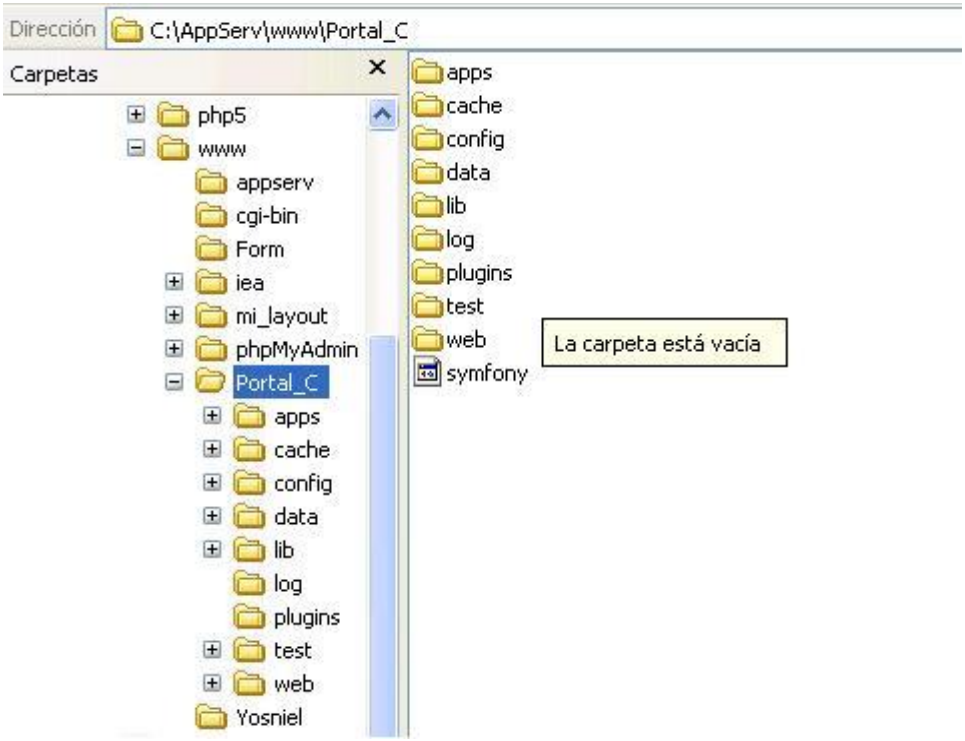

Figura # 3 Estructura de un proyecto en Symfony.

2.6 Patrones de Symfony.

Symfony está basado en el patrón MVC (Modelo Vista Controlador), este tiene como objetivo separar las funciones básicas de una aplicación web en tres niveles:

El modelo: representa la información con que trabaja la aplicación, la lógica de negocio.
La vista: se encarga de mostrar al usuario la información (Interfaz), normalmente como lenguaje HTML.
El controlador: procesa las interacciones del usuario y lleva a cabo los cambios en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Un ejemplo sería: si una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica

relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (8)

La siguiente figura ilustra el funcionamiento del patrón MVC.

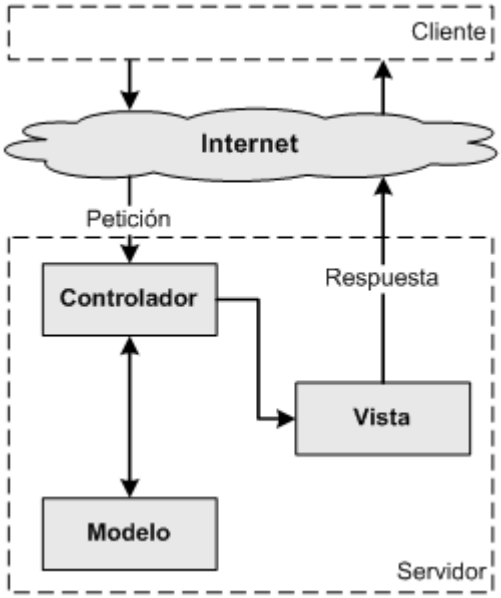

Figura # 4 Patrón MVC.

Además Symfony emplea los principales patrones Grasp: Creador, Experto, Alta Cohesión, Bajo Acoplamiento y Controlador, emplea también Patrones GOF, de las categorías Creacionales y Estructurales, de los primeros tenemos Singleton y Abstract Factory y por los últimos Decorator y Composite.

2.7 Estructura del portal.

El portal muestra un menú para acceder a las diferentes páginas con que cuenta, identificándose en esta sección vínculos a Servicios: donde se encuentran archivos relacionados con el tema de la calidad, los cuales el usuario podrá descargar para trabajar con ellos, Quiénes Somos: donde se presenta información del personal implicado en este portal, así como integrantes del proyecto, un Foro: donde es necesario estar autenticado para participar en las discusiones que se realicen, Áreas: aquí se puede acceder a los módulos donde se gestiona la información relacionada con los proyectos y los procesos de auditorías y revisiones, también el personal relacionado con las pruebas y los proyectos y las máquinas disponibles

por laboratorios, y es necesario estar logueado y tener credenciales de administrador o asesor al igual que para acceder a la página Administrar: en la cual se manejan los roles de los usuarios, las noticias entre otros aspectos administrativos del sitio. En la parte derecha de la página se muestra el formulario de autenticación, una sección de noticias relevantes, enlaces de interés para el navegante. La página principal o portada contendrá las noticias actuales, que tratarán tanto de la informática y la calidad como de otros ámbitos importantes.

2.8 Algoritmos

A continuación se realiza la descripción de los algoritmos principales para el funcionamiento correcto de la aplicación que se desarrolla.

2.8.1 Algoritmo utilizado para autenticar a los usuarios.

2.8.1.1 Descripción de los pasos.

1. El usuario accede al portal.
2. Se dirige a la sección de autenticación e introduce sus datos.
3. Presiona el botón enviar y se envían los datos hacia el Action (controlador).
4. En el Action se validan los datos y se invoca el método **Autenticar Usuario**.
5. El usuario es autenticado si los datos son los correctos y redireccionado a la página inicio con un mensaje de bienvenida.
6. Si los datos no son los correctos, se redirecciona a la página autenticarse, para introducir los datos nuevamente.

2.8.2 Registrar Nuevo Usuario.

2.8.2.1 Descripción de los pasos.

1. El usuario accede al portal.
2. El usuario se dirige a la sección de autenticación y presiona el vínculo *Registrarse Aquí*.
3. El sistema muestra un formulario para introducir los datos
4. Introduce sus datos, presiona el botón enviar y se envían los datos hacia el Action (controlador).
6. En el Action se validan los datos y se invoca el método **Registrar Usuario**.

7. El nuevo usuario es insertado en la base de datos si todos los datos requeridos son enviados y se muestra un mensaje indicando que la operación ha sido satisfactoria.
8. Si los datos son incorrectos se redirecciona nuevamente para el formulario.

2.8.3 Registrar Noticia.

2.8.3.1 Descripción de los pasos.

1. El usuario accede a la página.
2. Se autentica de forma satisfactoria
3. Accede al vínculo Registrar Noticia, y se muestra un formulario si es administrador.
4. El usuario llena los campos del formulario y envía los datos.
5. Los datos son validados y si son correctos se registra la nueva noticia en la base de datos.
6. Si los datos son incorrectos se redirecciona para el formulario anterior.
7. Si el usuario no es Administrador se muestra un mensaje indicando que no tiene privilegios para realizar esta acción.

2.8.4 Gestionar Estudiante.

2.8.4.1 Insertar estudiante.

Descripción de los pasos.

1. El usuario accede a la aplicación.
2. Se autentica satisfactoriamente.
2. Presiona el vínculo Registrar Estudiante y se muestra el formulario si es Administrador o Asesor del módulo.
3. El usuario introduce los datos de todos los campos del formulario y da clic en el botón enviar.
4. Los datos se validan y se llama al método **Insertar Estudiante**.
5. El nuevo estudiante es insertado en la base de datos.
6. Si los datos no están completos se redirecciona para el formulario anterior.

2.8.4.2 Actualizar estudiante.

Descripción de los pasos.

1. El usuario accede a la aplicación.

2. Se autentica satisfactoriamente.

2. Pincha en el vínculo Actualizar Estudiante y se muestra el formulario si es Administrador o Asesor del módulo.

3. El usuario introduce los datos de todos los campos del formulario y da clic en el botón enviar.

4. Los datos se validan y se llama al método **Actualizar Estudiante**.

5. El estudiante es actualizado en la base de datos.

6. Si los datos no están completos se redirecciona para el formulario anterior.

2.8.4.3 Eliminar estudiante.

Descripción de los pasos.

1. El usuario accede a la aplicación.

2. Se autentica satisfactoriamente.

2. Pincha en el vínculo Eliminar Estudiante y se muestra el formulario si es Administrador o Asesor del módulo.

3. El usuario escoge el estudiante a eliminar y da clic en el botón enviar.

4. Los datos son capturados y se llama al método **Eliminar Estudiante**.

5. El estudiante es eliminado de la base de datos.

2.8.5 Algoritmo Gestionar una Planilla.

2.8.5.1 Insertar Planillas.

Descripción de los pasos.

1. El usuario accede al portal.

2. Se dirige al módulo Auditorías y Revisiones.

3. El sistema muestra un listado de Paquetes de Revisiones, se presiona el botón insertar que aparece al final.

4. El sistema muestra un formulario donde se inserta el nombre del proyecto al cual se le realiza la revisión, se introduce el dato y se presiona el botón adicionar.

5. El sistema muestra un formulario con todas las planillas a insertar.

6. El usuario escoge la planilla e introduce los datos que se piden, luego presiona el botón adicionar y envía los datos.

7. Los datos son validados y se llama a los métodos **Insertar** correspondientes.
8. Si los datos no son correctos, se redirecciona al formulario anterior.

2.8.5.2 Actualizar Planillas.

Descripción de los pasos.

1. El usuario accede al portal.
2. Se dirige al módulo Auditorías y Revisiones.
3. El sistema muestra un listado de Paquetes de Revisiones, se da clic en el vínculo modificar al lado del paquete deseado.
5. El sistema muestra un formulario con todas las planillas del paquete que se quiere modificar.
6. El usuario escoge la planilla e introduce los datos que se piden, luego presiona el botón modificar y envía los datos.
7. Los datos son validados y se llama a los métodos **Actualizar** correspondientes.
8. Si los datos no son correctos, se redirecciona al formulario anterior.

2.9 Conclusiones Parciales.

En el capítulo se analiza el diseño obtenido de la fase anterior, para proceder con la fase de implementación, se aborda sobre el estándar de codificación realizado para basarse a la hora de escribir el código de la aplicación, se muestra el diagrama de componentes realizado y se presenta la estructura que proporciona Symfony a la aplicación. Se exponen además los principales algoritmos realizados con el fin de obtener una aplicación robusta y correcta.

CAPÍTULO 3: Pruebas del Software.

3.1 Introducción

En este apartado se realiza una investigación sobre las pruebas necesarias para lograr la calidad de un software, la importancia de estas en todo el proceso de desarrollo de la aplicación y los tipos que existen. Se diseñan los casos de prueba para comprobar la aplicación realizada, con vistas a minimizar los errores y que se pueda obtener un sistema completamente funcional, acorde a lo que quiere el cliente. Se lleva a cabo un análisis de los resultados obtenidos en las pruebas y se procede con la corrección de defectos en caso de que se detecte alguno, al culminar esta última parte se prueba nuevamente el software, y así sucesivamente hasta que el producto quede libre de defectos y errores.

3.2 Pruebas de Software.

Probar es una actividad fundamental en el proceso de desarrollo de sistemas de software. Esta disciplina ha surgido a raíz de la fuerte demanda de aplicaciones en todos los procesos que se desarrollan en la actualidad. El aseguramiento de la calidad tiene en cuenta todas aquellas acciones planificadas y sistemáticas que son necesarias para propiciar la seguridad de que un producto o servicio satisfaga los requisitos de calidad establecidos. Para ello es necesario realizar distintas pruebas a un determinado software.

Una vez que se ha generado el código comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos. (7)

La calidad de un software está determinada, entre otras cosas, por la concordancia entre los requisitos especificados por el cliente y lo que se programa por parte de los implementadores. Para comprobar el grado en que se ha cumplido con tales requisitos se llevan a cabo las pruebas de software, las cuales definen un conjunto de actividades de verificación que abarcan las características que determinan la calidad de un software. Para lograr la calidad es necesario comprobar las funcionalidades diseñando casos de prueba que definen como proceder. Las pruebas deben aplicarse durante todo el ciclo de vida del software, dedicándole gran parte del esfuerzo total de desarrollo, estas deben planificarse

correctamente desde el inicio y establecer que se va a hacer, cómo y quién, y en qué condiciones. Es bueno que los desarrolladores prueben su producto, pero se recomienda que una tercera persona, que no haya intervenido en el producto examine o pruebe el software, estos pueden descubrir mayor cantidad de errores o fallos y de manera más fácil.
Se propone escoger tipos de prueba que se adapten mejor al software que se prueba, teniendo en cuenta el lenguaje de programación, las características de los desarrolladores, la plataforma que se utiliza, el tipo de aplicación (escritorio o web), el proceso de desarrollo, si realiza conexiones a base de datos, los errores, entre otras observaciones.

La prueba es un proceso de ejecución de un programa con la intención de descubrir errores, tienen éxito si descubren alguno, y fracasan si los defectos existen, pero estas no los encuentran. **(8)**

3.3 Tipos de prueba.

Cualquier producto, ya sea del ámbito informático o no, puede probarse de las dos siguientes formas: (a) conociendo la función específica para la que fue diseñado el mismo, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y, al mismo tiempo buscando errores en cada función. **(8)**
(b), conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que todas las pruebas encajan, es decir, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.
El primer enfoque de prueba recibe el nombre de prueba de caja negra, el segundo prueba de caja blanca.

- Las pruebas de **Caja Blanca** (conocidas también como *pruebas de **caja de cristal***), es un método de diseño de caso de prueba que usa la estructura de control del diseño procedimental para obtener los casos de uso. En esta se debe verificar el funcionamiento interno de la aplicación comprobando que esté acorde a lo especificado, y que los componentes internos funcionen correctamente, por lo que se desarrolla en base a los caminos lógicos del software. Esta prueba es la que mostrará de forma más clara los principales problemas.

Las pruebas de Caja Blanca deben verificar que:

- Los caminos mínimos de cada módulo se ejecuten al menos una vez.

- Todas las decisiones lógicas se prueben en sus ramas verdaderas y falsas.
- Sean ejecutados todos los bucles o ciclos con los límites que se les haya definido.
- Sean ejecutadas las estructuras internas de datos para asegurar su validez.

Métodos o herramientas de Caja Blanca:

- Prueba del camino básico.
- Pruebas de bucles.
- Pruebas de condiciones.
- Pruebas de declaración.
- Múltiples Condiciones.
- Pruebas de decisión

➢ Las pruebas de **Caja Negra (caja opaca)**, son aquellas en las que conociendo las funciones del programa, se debe demostrar que estas son satisfactorias, por lo que solo se realiza sobre las interfaces de dicho software. Se centran principalmente en los requisitos funcionales del sistema.

Permiten encontrar:

✓ Funciones incorrectas o ausentes.
✓ Errores de interfaz.
✓ Errores en estructuras de datos o en accesos a las Bases de Datos externas.
✓ Errores de rendimiento.
✓ Errores de inicialización y terminación.

Técnicas o pruebas de Caja Negra

Estas son complementarias a las de caja blanca, pero en la práctica solo se realizan normalmente las de caja negra.

- Análisis de valores límites.
- Partición Equivalente.
- Pruebas de transición de estado.
- Grafo causa efecto.
- Pruebas de tabla de decisión.

- Árbol de clasificación.
- Arreglos ortogonales.
- Casos de prueba.

Comúnmente las pruebas que se le practican a las aplicaciones o software que se crean son las de Caja Negra, para comprobar que el programa tiene las funcionalidades implementadas correctamente, y que realiza lo que se espera de él, también para verificar que no se haya quedado ninguna funcionalidad sin resolver o implementar. Estas pruebas encuentran además errores que podrían llegar a ser vistos por los usuarios, como lo son las faltas de ortografía, los vínculos incorrectos, imágenes que no existen o que no se muestran, entre otros.

Se necesita realizar la mayor cantidad de pruebas posibles para dar de alta a un producto informático o programa, por tanto existen otras pruebas que son necesarias practicar, entre ellas están las pruebas de integración, del sistema, de carga, de validación, de verificación, de unidad, de desempeño, de documentación, de regresión, de estrés (stress), así como las de aceptación, entre otras.

3.4 Diseño de los Casos de Pruebas.

Conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Siempre es ejecutada como una unidad, desde el comienzo hasta el final. **(8)**

Debe verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Un caso de prueba es aquel que tiene una alta probabilidad de descubrir un defecto no detectado hasta entonces.

Para comprobar el correcto funcionamiento de los distintos módulos de la aplicación se realizó el diseño de casos de pruebas de software a cada caso de uso, tratando de que estos tengan una alta probabilidad de encontrar errores, para conseguir esto se aplican las técnicas de prueba de Caja Negra.

Se presenta a continuación la tabla con los resultados obtenidos luego de realizar las pruebas a los módulos de RRHH y AR.

Tabla 2 Resultado de la pruebas.

Casos de Uso Críticos	Resultado del sistema	Resultado
Gestionar Estudiante	Inserta, elimina o actualiza un estudiante.	2 No Conformidades Encontradas
Gestionar Profesor	Inserta, elimina o actualiza un profesor.	1 No Conformidad Encontrada
Gestionar Computadora	Inserta, elimina o actualiza una computadora.	Satisfactorio
Mostrar Reporte	Muestra un listado de todos los recursos humanos existentes.	Satisfactorio
Gestionar paquete de Revisiones	Inserta, elimina o actualiza una revisión y las planillas presentes en ellas.	4 No Conformidades Encontradas

Tabla que muestra las variables de entrada del caso de prueba Gestionar Registro de Evaluaciones del módulo Auditorías y Revisiones.

Tabla 3 Variables de entrada del caso de prueba Gestionar Registro de Evaluaciones

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Número de Control.	Campo de texto.	No	Este campo debe contener números enteros.
2	Clave del plan de evaluaciones.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
3	Nombre del Revisor.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
4	Rol del Revisor.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
5	Jefe del Proyecto.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
6	Total de NC.	Campo de texto.	No	En este campo solo puede introducir números enteros.
7	Observaciones.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
8	Fecha de evaluación.	Campo de selección.	No	Se debe seleccionar una fecha.

Tabla que muestra las variables de entrada del caso de prueba Gestionar Profesor del módulo de Recursos Humanos.

Tabla 4 Variables de entrada del caso de prueba Gestionar Profesor

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre del profesor.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
2	Apellidos del estudiante.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
3	Solapín.	Campo de texto.	No	En este campo se pueden introducir números y letras.
4	Correo.	Campo de texto.	No	Este debe contener números.
5	Número de computadora.	Campo de texto.	No	Este debe contener números.
6	Apartamento.	Campo de texto.	No	Este debe contener números.
7	Teléfono.	Campo de texto.	No	Este debe contener números.
8	Rol	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
9	Estudiante que tutorea.	Campo de texto.	No	En este campo se puede introducir cualquier tipo de dato.
10	Carnet de Identidad.	Campo de texto.	No	Este debe contener números.
11	¿Es tutor?	Campo de texto.	No	Este debe contener números.

Tabla 5 No conformidades encontradas.

# N C	No Conformidad	Importancia	Estado	Módulo
1	No se actualizan dos campos en la tabla de estudiantes.	Media	Pendiente	Recursos Humanos
2	No se actualizan dos campos en la tabla de profesores.	Media	Pendiente	Recursos Humanos
3	En la página Actualizar Profesor, se muestra un error en el estudiante que tutorea.	Media	Pendiente	Recursos Humanos
4	La acción Registrar Reunión de Apertura no funciona correctamente.	Alta	Pendiente	Auditorías y Revisiones
5	La acción Registrar Reunión de Cierre no funciona correctamente.	Alta	Pendiente	Auditorías y Revisiones
6	La acción Actualizar Reunión de Apertura no funciona correctamente.	Alta	Pendiente	Auditorías y Revisiones
7	La acción Actualizar Reunión de Cierre no funciona correctamente.	Alta	Pendiente	Auditorías y Revisiones

3.5 Análisis de las no conformidades encontradas.

Las no conformidades encontradas, tuvieron mayor peso en el módulo de auditorías y revisiones, ya que fue aquí donde se detectaron mayor cantidad y todas con importancia alta, como se pudo ver en la *tabla 3*, por otra parte las que se detectaron en el módulo de recursos humanos tenían importancia media, ya se trabaja en la corrección de estas y luego serán corregidas las del otro módulo. Después de concluirse la fase de erradicación de los defectos encontrados, se ejecutará una segunda ronda de pruebas al software verificando que esté libre de defectos.

3.6 Conclusiones Parciales.

En el capítulo se estudian las pruebas que se realizan para garantizar el buen funcionamiento de un software. Se discuten los casos de prueba diseñados y que se le practicaron a la aplicación Portal de Calidad de la Facultad 9, con vías a garantizar que dicho software cumpla con los requerimientos planteados por el cliente. Los resultados arrojados fueron satisfactorios para la aplicación, ya que solo se encontraron 7 no conformidades las cuales fueron corregidas, realizándose posteriormente una segunda fase de pruebas, en la cual no se encontraron defectos y aunque una prueba no garantiza la ausencia de estos, se puede decir que la aplicación es segura, confiable y cumple con los requerimientos y funcionalidades especificadas, por lo que está apta para ser publicada y brindar servicios a la comunidad universitaria.

Conclusiones Generales.

Como parte de la investigación se estudiaron otros sistemas tanto de la Universidad como del exterior, los cuales no respondían a todas las necesidades que se requerían, pero sirvieron de apoyo para la confección de este. Se analizaron los métodos, herramientas o materiales más óptimos para lograr un software con la calidad que se necesita.

Para el momento en que se termina el actual trabajo o investigación, se puede afirmar que con su desarrollo se han cumplido los objetivos trazados, gestionando la información de los módulos de Recursos Humanos y Auditorías y Revisiones, lo que permitirá realizar las actividades necesarias de forma ágil y en el tiempo previsto, y que puedan ser controladas en su transcurso o una vez terminadas. Se pone a disposición de los estudiantes y demás trabajadores una serie de documentos que ayudan o apoyan en la tarea de asegurar la calidad de un producto, y se habilita un foro donde se debatirá sobre estos temas, permitiendo interactuar con los profesores capacitados en este ámbito, entre otras funciones que el usuario podrá realizar una vez iniciada la aplicación.

Se probaron ambos módulos una vez culminada la implementación, para verificar el correcto funcionamiento de cada uno en los puntos más críticos, finalizando con esta actividad el presente trabajo.

Con la terminación de este software se da un paso más en la informatización de los distintos procesos de la sociedad, demostrando que muchas de las actividades pueden computarizadas, lo que traería consigo mayor calidad, y desarrollo en menor tiempo, facilitándole el trabajo al hombre.

Recomendaciones

Se recomienda primeramente, que se le dé continuación a este trabajo, que se mantenga actualizado el portal, que se publiquen diariamente las noticias más relevantes en el ámbito de la informática, que se le agregue un buscador y que se continúe enriqueciendo el mismo, para propiciar una mejor gestión de la información y brindar un mejor servicio a los usuarios, así como que sea publicado para beneficio de la comunidad universitaria y los grupos de calidad, sobre todo el de la Facultad 9.

Que se le realicen pruebas de Caja Blanca, para verificar el funcionamiento interno de la Aplicación, y que se le agregue autenticación por el dominio UCI.

Que la investigación sea consultada por aquellas personas que investiguen sobre el tema, visitando la bibliografía que se presenta para mayor información.

Referencias Bibliográficas

1. Romero, Arturo Luis y Miranda, Sandor Luis. GestioPolis. *GestioPolis.* [En línea] 10 de agosto de 2007. [Citado el: 11 de diciembre de 2009.] http://www.gestiopolis.com/administracion-estrategia/la-calidad-historia-conceptos-y-terminos-asociados.htm.

2. Nistal Rosique, Gloria. Asociación de Técnicos de Informática (ATI). *Asociación de Técnicos de Informática (ATI).* [En línea] 1999. [Citado el: 11 de diciembre de 2009.] http://www.ati.es/novatica/1999/137/pres137.html.

3. Guglielmetti, Marcos. MasterMagazine. *MasterMagazine.* [En línea] 2004. [Citado el: 14 de enero de 2010.] http://www.mastermagazine.info/termino/6349.php.

4. Vegas, Jesús. Dpto Inform. Unic Valladolid. *Dpto Inform. Unic Valladolid.* [En línea] 21 de 03 de 2002. [Citado el: 22 de enero de 2010.] http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html.

5. Félix, Alvaro del Castillo San. Entorno de acs. *Entorno de acs.* [En línea] 2000. [Citado el: 19 de enero de 2010.] http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html.

6. Eguíluz, Javier. Symfony. *Symfony.* [En línea] 2010. [Citado el: 25 de enero de 2010.] http://www.symfony.es

7. Pressman, Roger. *Ingeniería del Software. Un enfoque práctico.* Madrid : Universidad Pontificia de Salamanca campus Madrid, 2001. Quinta Edición.

8. Zaninotto, François y Potencier, Fabien. Librosweb. *Symfony 1.0, la guía definitiva.* [En línea] [Citado el: 14 de junio de 2010.] http://www.librosweb.es/symfony_1_0/capitulo2/el_patron_mvc.html.

Bibliografías

Milenium. *Milenium*. [En línea] 2009. [Citado el: 23 de enero de 2010.]

http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm

Andrés Villanueva Manjarres. Desarrollando Web. *Desarrollando Web*. [En línea] 2007. [Citado el: 14 de enero de 2010.]
http://www.desarrollandoweb.com/internet/tipos-de-portales.php

Miguel Angel Alvarez. Desarrollo Web. *Desarrollo Web*. [En línea] 04 de junio de 2003. [Citado el: 16 de enero de 2010.] http://www.desarrolloweb.com/articulos/1178.php

Rafael Piñero. Visual Beta. *Visual Beta*. [En línea] 11 de marzo de 2009. [Citado el: 16 de enero de 2010.] http://www.visualbeta.es/8634/software-libre/aptana-studio-un-entorno-de-desarrollo-completo/

Taringa. *Taringa*. [En línea] 11 de febrero de 2009. [Citado el: 19 de enero de 2010.]
http://www.taringa.net/posts/ebooks-tutoriales/2147867/Aptana-Studio-(Uno-de-los-mejores-programas-de-programacion).html

Alvaro del Castillo San Félix. ACS. *ACS*. [En línea] 11 de mayo de 2005. [Citado el: 20 de enero de 2010.] http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/c20.html

Hispavista Software. *Hispavista Software.* [En línea] 01 de enero de 1998. [Citado el: 21 de enero de 2010.] http://software.hispavista.com/s/Xitami+Multithreaded+Webserver

Forum Wiaderko. *Forum Wiaderko.* [En línea] 24 de febrero de 2009. [Citado el: 23 de enero de 2010.]
http://www.forum.wiaderko.com/es/305758-post1.html

Miguel Angel Alvarez. Desarrollo Web. *Desarrollo Web.* [En línea] 09 de mayo de 2001. [Citado el: 20 de enero de 2010.] http://www.desarrolloweb.com/articulos/392.php

Miguel Angel Alvarez. Desarrollo Web. *Desarrollo Web*. [En línea] 08 de julio 2002. [Citado el: 21 de enero de 2010.] http://www.desarrolloweb.com/articulos/831.php

BIBLIOGRAFÍAS

Miguel Angel Alvarez. Desarrollo Web. *Desarrollo Web*. [En línea] 09 de julio de 2002. [Citado el: 21 de enero de 2010.] http://www.desarrolloweb.com/articulos/832.php

aNieto2k. *aNieto2k.* [En línea] Junio de 2007. [Citado el: 12 de enero de 2010.] http://www.anieto2k.com/2007/06/06/y-erase-una-vez-javascript/

Mariano Amartino. Uberbin. *Uberbin.* [En línea] 28 de marzo de 2005. [Citado el: 16 de enero de 2010.] http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php

Masadelante. *Masadelante.* [En línea]. [Citado el: 21 de enero de 2010.] http://www.masadelante.com/faqs/base-de-datos

Instituto de Tecnología Massachusetts. OpenCourseWare. *OpenCourseWare*. [En línea] 2003. [Citado el: 23 de enero de 2010.] http://mit.ocw.universia.net/curso11208/11/11.208/IAP02/lecture-notes/lecture2-2.html

Lauro Soto. Mitecnologico. *Mitecnologico*. [En linea]. [Citado el: 20 de enero de 2010.] http://www.mitecnologico.com/Main/HerramientasDeBasesDeDatos

Miguel Angel Alvarez. Desarrollo Web. *Desarrollo Web.* [En línea]. [Citado el: 14 de enero de 2010.] http://www.desarrolloweb.com/directorio/bases_de_datos/mysql/

Daniel Pecos. NetPecos. *NetPecos.* [En línea] 07 de junio de 2002. [Citado el: 19 de enero de 2010.] http://www.netpecos.org/docs/mysql_postgres/x57.html

Maestros del web. *Maestros del Web.* [En línea] 31 de julio de 2007. [Citado el: 05 de febrero de 2010.] http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/

Javier Eguíluz. Slide Share Inc. *Slide Share*. [En línea] 2008. [Citado el: 02 de febrero de 2010.] http://www.slideshare.net/javier.eguiluz/curso-symfony-clase-1

Libros Web. *Libros Web.* [En línea] 25 de agosto de 2009. [Citado el: 12 de enero de 2010.] http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html

BIBLIOGRAFÍAS

WopSolutions. *WopSolutions.* [En línea] 2009. [Citado el: 15 de enero de 2010.] http://www.wopsolutions.com/ventajas-de-las-aplicaciones-web.html

XOR Consultores C.A. *XOR Consultores C.A.* [En línea] 2007. [Citado el: 21 de enero de 2010.] http://www.xorcon.com/servicio.php

Esencia Humana. *Esencia Humana.* [En línea] 2008. [Citado el: 04 de febrero de 2010.] http://www.esenciahumana.com.mx/Servicios/AplicacionesWeb/VentajasBeneficiosAplicaciones.html

Andrés Villanueva Manjarres. DesarrollandoWeb. *DesarrollandoWeb.* [En línea] 2008. [Citado el: 13 de enero de 2010.] http://www.desarrollandoweb.com/internet/tipos-de-portales.php

Webexperto. *Webexperto.* [En línea] 16 de junio de 2006. [Citado el: 2 de febrero de 2010.] http://www.webexperto.com/downloads/83/xitami-25c2/

Gómez, Ramírez, De Villa. EAFIT. *EAFIT.* [En línea] 15 de septiembre de 2005. [Citado el: 8 de enero de 2010.] http://www.eafit.edu.co/NR/rdonlyres/E12D52F4-E868-4C79-98C2-B917EADA7892/0/Frameworks.pdf

Marcus Eduardo Markiewicz, Carlos J.P. de Lucena. Association for Computing Machinery. *ACM*. [En línea] Julio-Agosto 2001. [Citado el: 09 de febrero de 2010.] http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html

Joaquín García. WebEstilo. *WebEstilo.* [En línea] 2004. [Citado el: 09 de enero de 2010.] http://www.webestilo.com/html/cap1a.phtml

Joaquín García. WebEstilo. *WebEstilo.* [En línea] 2004. [Citado el: 09 de enero de 2010.] http://www.webestilo.com/html/cap2a.phtml

HTMLPOINT. *HTMLPOINT.* [En línea] 2006. [Citado el: 10 de febrero de 2010.] http://www.htmlpoint.com/iis/01.htm

Microsoft. *Microsoft.* [En línea] 11 de julio de 2007. Citado el: 10 de enero de 2010.] http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.mspx

BIBLIOGRAFÍAS

CrearGratisUnaPaginaWeb. *CrearGratisUnaPaginaWeb.* [En línea] 14 de julio 2009. [Citado el: 10 de febrero de 2010.] http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/

Maestros del Web. *Maestros del Web.* [En línea] 03 de julio de 2007. [Citado el: 10 de enero de 2010.] http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/

Nelson Druell. E-articles. *E-articles.* [En línea] 2005. [Citado el: 16 de enero de 2010.] http://e-articles.info/t/i/169/l/es/

Cristian Van Der Henst S. Maestros del Web. *Maestros del Web.* [En línea] 23 de abril de 2001. [Citado el: 22 de enero de 2010.] http://www.maestrosdelweb.com/editorial/aspintro/

Javier J Gutiérrez**.** Lenguaje y Sistemas Informáticos. Universidad de Sevilla. [En línea] 2008. [Citado el: 04 de marzo de 2010.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.

Alejandro Pérez García. Desarrollo Web. *Desarrollo Web.* [En línea] 21 de febrero de 2006. [Citado el: 04 de marzo de 2010.] http://www.desarrolloweb.com/articulos/2380.php

Seam City. Seam City. [En línea] 7 de enero de 2008. [Citado el: 04 de marzo de 2010.] http://seamcity.madeinxpain.com/archives/comparativa-cocoon-2

Sscript. *Sscript.* [En línea] 09 de marzo de 2002. [Citado el: 04 de marzo de 2010.] http://sscript.infdj.com/Varios/xitami.htm

CarlosWeb`s Weblog .CarlosWeb`s Weblog. [En línea] 19 de diciembre de 2008. [Citado el: 08 de marzo de 2010.] http://carlosweb2.wordpress.com/2008/12/19/jsf/

Miguel Angel Alvarez. Desarrollo Web. [En línea] 23 de noviembre de 2009. [Citado el: 26 de mayo de 2010.] http://www.desarrolloweb.com/articulos/codeigniter.html

Juan Miguel Calcaño. ATINOX.NET. [En línea] 07 de mayo de 2009. [Citado el: 26 de mayo de 2010.] http://www.asinox.net/2009/05/07/introduccion-a-codeigniter/

BIBLIOGRAFÍAS

Tufuncion. Tufuncion. [En línea] 30 de enero de 2007. [Citado el: 26 de mayo de 2010.] http://www.tufuncion.com/cakephp

Book.Cakephp. Book.Cakephp. [En línea]. [Citado el: 26 de mayo de 2010.] http://book.cakephp.org/es/compare/13/Basic-Principles-of-CakePHP

Autor. Sitio. *Nombre del Sitio*. [En línea] 2009. [Citado el: 22 de febrero de 2010.] http://chileforge.cl/docman/view.php/81/108/estandar_de_codificacion.pdf

Carlos Fernández Guillort. *Universidad Jesuita de Guadalajara*. [En línea]. [Citado el: 22 de febrero de 2010.] http://progra.iteso.mx/estandares/estandar%20codificacion%20c++/estandarcodificacion.pdf

Dirección General de Gobierno Digital. Chubut. [En línea] 2009. [Citado el: 22 de febrero de 2010.] http://www.chubut.gov.ar/informatica/docs/EstandaresCodificacion.pdf

UADE. *Universidad Autónoma de España*. [En línea] 2009. [Citado el: 21 de abril del 2010.] http://my.opera.com/pelican0/blog/show.dml/564829

Pruebas de Software. [En línea] 2009. [Citado el: 29 de abril del 2010.] http://www.pruebasdesoftware.com/pruebadeaceptacion.htm

Thomas Rabaix. Grupos Google. [En línea] 18 de mayo de 2008. [Citado el: 14 de junio del 2010.] http://groups.google.com/group/symfony-users/msg/cd94d2ddb2057355

Javier Eguiluz. Symfony. [En línea] 10 de marzo de 2007. [Citado el: 14 de junio del 2010.] http://www.symfony.es/2007/03/10/patron-composite-en-php-y-symfony/

Alberto Marín Narvaez. Albertomarín62.blogspot. [En línea] 20 de mayo de 2008. [Citado el: 14 de junio del 2010.] http://albertomarin62.blogspot.com/2008/05/sobre-symfony.html

Anexos.

1. Autenticar usuario.

```
public function executeAutenticar(sfWebRequest $request)
{
        $this->otros = Doctrine::getTable('Usuario')
                    ->createQuery('h')
                    ->execute();

 if ($this->getRequest()->getMethod() == sfRequest::POST)
        {
        $user = $this->getRequestParameter('user');
        $pass = md5($this->getRequestParameter('pass'));
        $si = false;
        $entro = false;
        foreach ($this->otros as $todo):
          {
          if($todo->getUsuario()==$user && $todo->getContrasenna() == $pass)
            {
             $rol = Doctrine::getTable('UserRol')
                    ->createQuery('b')
                    ->where('b.solapin= ?', $todo->getSolapin())
                    ->execute();

              if (!$this->getUser()->isAuthenticated())
                {
                  $this->getUser()->setAuthenticated(true);
                  $this->getUser()->setAttribute('nombre', $todo->getNombre());
                  $this->getUser()->setAttribute('apellidos', $todo->getApellidos());
                  $this->getUser()->addCredential($rol[0]->getIdrol());
                  $entro = true;
                }
```

Figura # 5 Algoritmo autenticar usuario-a.

```
            }
          }
        endforeach;
      }
      if(!$entro)
        {
          $this->getUser()->setFlash('error', sprintf('Datos incorrectos'));
          $this->redirect('inicio/reg_user');
        }
        else
          {
            $this->getUser()->setFlash('notice', sprintf('Autenticado correctamente'));
            $this->redirect('inicio/index');
          }
}
```

Figura # 6 Algoritmo autenticar usuario-b.

2. Insertar estudiante.

```
public function executeReg_estud()
{
   if ($this->getRequest()->getMethod() == sfRequest::POST)
      {
          $estudiante = Doctrine::getTable('Estudiante')
                        ->createQuery('a')
                        ->where('a.Solapin = ?', $this->getRequestParameter('sol'))
                        ->execute();

          if($estudiante[0]->getSolapin() != null) //Si esta el solapin muestro un mensaje de error.
          {
            $this->getUser()->setFlash('error', sprintf('El estudiante "'.
            $this->getRequestParameter('sol').'" ya se encuentra registrado en la base de datos,
                                                    inserte otro por favor'));
            $this->redirect('recursos_humanos/reg_estudiante');
          }
           $ci = $this->getRequestParameter('ci');
           $nom  = $this->getRequestParameter('nom');
           $ape = $this->getRequestParameter('ape');
           $nota = $this->getRequestParameter('nota');
           $sol = $this->getRequestParameter('sol');
           $correo = $this->getRequestParameter('correo');
           $grupo = $this->getRequestParameter('grupo');
           $tema = $this->getRequestParameter('tema');
           $anno = $this->getRequestParameter('anno');
           $apto = $this->getRequestParameter('apto');
           $telef = $this->getRequestParameter('telef');
           $rolc = $this->getRequestParameter('rol');
       $area_t = $this->getRequestParameter('area');
           $lab = $this->getRequestParameter('lab');
       $num = $this->getRequestParameter('num');
```

Figura # 7 Algoritmo insertar estudiante-a.

```
$comp = Doctrine::getTable('Computadora')
            ->createQuery('a')
            ->where('a.num = ?', $this->getRequestParameter('num'))
            ->andWhere('a.lab = ?', $this->getRequestParameter('lab'))
            ->execute();
$recurso = Doctrine::getTable('RecursosHumanos')
            ->createQuery('a')
            ->where('a.solapin = ?', $this->getRequestParameter('tutor'))
            ->execute();

$tutor = $recurso[0]->getNombre();

if($comp[0]->getIdpc() == null) //Si no está la computadora, la adiciono.
  {
     ComputadoraTable::Insertar_C($this->getRequestParameter('lab'),
            $this->getRequestParameter('num'));
  }
    $comp = Doctrine::getTable('Computadora')
            ->createQuery('a')
            ->where('a.num = ?', $this->getRequestParameter('num'))
            ->andWhere('a.lab = ?', $this->getRequestParameter('lab'))
            ->execute();
    $id_comp = $comp[0]->getIdpc();

    RecursosHumanosTable::Insertar_E($nom,$ape, $sol, $correo, $apto,
            $telef, $area_t, $rolc, $id_comp);

    EstudianteTable::Insertar_E($sol, $grupo, $anno, $nota, $tutor);
    $this->getUser()->setFlash('notice', sprintf('El estudiante "'.
    $this->getRequestParameter('ci').'" fue insertado correctamente'));
    $this->redirect('recursos_humanos/lis_rec_est');
```

Figura # 8 Algoritmo insertar estudiante-b.

3. Actualizar estudiante.

```
public function executeAct_estudiante()
{
  if ($this->getRequest()->getMethod() == sfRequest::POST)
      {
      $soll = $this->getRequestParameter('soll');
      $sola = $this->getRequestParameter('sola');
      $estudiante = Doctrine::getTable('Estudiante')
          ->createQuery('a')
          ->where('a.solapin = ?', $soll)
          ->execute();
      $estudiante2 = Doctrine::getTable('Estudiante')
          ->createQuery('a')
          ->where('a.solapin = ?', $sola)
          ->execute();
      $recurso = Doctrine::getTable('RecursosHumanos')
          ->createQuery('a')
          ->where('a.solapin = ?', $this->getRequestParameter('soll'))
          ->execute();
       if($this->getRequestParameter('sola') != $soll) //pregunto si son diferentes.
          {
              if($estudiante2[0]->solapin == null) //si no esta el nuevo lo actualizo
              {
                  $estudiante->delete();
                  $recurso[0]->solapin = $sola;
              }
          }
      $estudiante[0]->grupo = $this->getRequestParameter('grupo');
      $estudiante[0]->anno = $this->getRequestParameter('anno');
      $estudiante[0]->nota_pp = $this->getRequestParameter('nota');
      $estudiante[0]->tutor = $this->getRequestParameter('tutor');
      $recurso[0]->nombre = $this->getRequestParameter('nom');
      $recurso[0]->telef = $this->getRequestParameter('telef');
```

Figura # 9 Algoritmo actualizar estudiante-a.

```
$recurso[0]->apto = $this->getRequestParameter('apto');
$recurso[0]->apellidos = $this->getRequestParameter('ape');
$recurso[0]->correo = $this->getRequestParameter('correo');
$recurso[0]->rolcalidad = $this->getRequestParameter('rol');
$recurso[0]->area_trab = $this->getRequestParameter('area');
    $comp = Doctrine::getTable('Computadora')
            ->createQuery('a')
            ->where('a.num = ?', $this->getRequestParameter('nume'))
            ->andWhere('a.lab = ?', $this->getRequestParameter('lab'))
            ->execute();
  if($nume != $comp[0]->getNum() || $this->getRequestParameter('lab') != $comp[0]->getNum())
  {
    $nume = $this->getRequestParameter('nume');
    if($comp[0]->getNum() == null) //Si no está la adiciono.
    {
       ComputadoraTable::Insertar_C($this->getRequestParameter('lab'), $nume);
       $comp2 = Doctrine::getTable('Computadora')
            ->createQuery('a')
            ->where('a.num = ?', $this->getRequestParameter('nume'))
            ->andWhere('a.lab = ?', $this->getRequestParameter('lab'))
            ->execute();
       $recurso[0]->idcomp= $comp2[0]->getIdpc();
    }
    else
        {
            $comp2 = Doctrine::getTable('Computadora')
            ->createQuery('a')
            ->where('a.num = ?', $this->getRequestParameter('nume'))
            ->andWhere('a.lab = ?', $this->getRequestParameter('lab'))
            ->execute();
            $recurso[0]->idcomp= $comp2[0]->getIdpc();
        }
  }
```

Figura # 10 Algoritmo actualizar estudiante-b.

```
        $recurso[0]->save();
        if($this->getRequestParameter('sola') != $soll) //Si no está la adiciono.
            {
                EstudianteTable::Insertar_E($sola, $this->getRequestParameter('grupo'),
                $this->getRequestParameter('anno'), $this->getRequestParameter('nota'),
                $this->getRequestParameter('tutor'));
            }
            else
                {
                    $estudiante[0]->save();
                }
        $this->getUser()->setFlash('notice', sprintf('El estudiante "'.
                $this->getRequestParameter('sola').'" fue actualizado correctamente'));
        $this->redirect('recursos_humanos/lis_rec_est');
        }
        else
         {
         $this->parasol = Doctrine::getTable('RecursosHumanos')
         ->createQuery('a')
         ->where('a.solapin = ?', $this->getRequestParameter('sol'))
         ->execute();
         $this->estud = Doctrine::getTable('Estudiante')
         ->createQuery('a')
         ->where('a.solapin = ?', $this->getRequestParameter('sol'))
         ->execute();
         $this->paracomp = Doctrine::getTable('Computadora')
         ->createQuery('a')
         ->where('a.idpc = ?', $this->parasol[0]->getIdcomp())
         ->execute();
         }
}
```

Figura # 11 Algoritmo actualizar estudiante-c.

4. Registrar Planilla de Acciones Correctivas.

```
public function executeReg_plan_acciones_correct() {
    if ($this->getRequest()->getMethod() == sfRequest::POST)
    {
        $fecha= $this->getRequestParameter('fecha');
        $num  = $this->getRequestParameter('num');
        $proy = $this->getRequestParameter('proy');
        $red = $this->getRequestParameter('red');
        $cargo = $this->getRequestParameter('cargo');
        $num_ac = $this->getRequestParameter('num_ac');
        $elem = $this->getRequestParameter('elem');
        $acc = $this->getRequestParameter('acc');

        if($this->getRequestParameter('imp') == 0) {
            $imp="Alta";
        }
        else if($this->getRequestParameter('imp') == 1) {
            $imp="Media";
        }
        else {
            $imp="Baja";
        }
        if($this->getRequestParameter('est') == 0) {
            $est="P";
        }
        else if($this->getRequestParameter('est') == 1) {
            $est="E";
        }
        else {
            $est="T";
        }
```

Figura # 12 Insertar planilla acciones correctivas-a.

```
        $nume = Doctrine::getTable('AccionesCorrectivas')
                ->createQuery('a')
                ->where('a.num_control = ?', $this->getRequestParameter('num'))
                ->execute();

        if($nume[0]->getNumControl() != null) //Si esta muestro el error.
        {
            $this->getUser()->setFlash('error', sprintf('La Planilla de acciones correctivas "'.
                    $this->getRequestParameter('num').'" ya existe'));
            $this->redirect('auditoria_rev/reg_plan_rev');
        }

        AccionesCorrectivasTable::Insertar_AccC($red, $proy, $fecha, $cargo, $num_ac,
        $elem, $acc, $imp, $est, $num);
        $this->getUser()->setFlash('notice', sprintf('La Planilla de acciones correctivas
        con numero:  "'.
        $this->getRequestParameter('num').'" se ha insertado correctamente'));
        $this->redirect('auditoria_rev/reg_plan_rev');
    }
}
```

Figura # 13 Insertar planilla acciones correctivas-a.

5. Actualizar Planilla de Acciones Correctivas.

```
public function executeAct_plan_acc_correct()
{
    if ($this->getRequest()->getMethod() == sfRequest::POST)
    {
        $fecha= $this->getRequestParameter('fecha');
        $num  = $this->getRequestParameter('num');
        $red = $this->getRequestParameter('red');
        $cargo = $this->getRequestParameter('cargo');
        $num_ac = $this->getRequestParameter('num_ac');
        $elem = $this->getRequestParameter('elem');
        $acc = $this->getRequestParameter('acc');

        if($this->getRequestParameter('imp') == 0) {
            $imp="Alta";
        }
        else if($this->getRequestParameter('imp') == 1) {
            $imp="Media";
        }
        else {
            $imp="Baja";
        }
        if($this->getRequestParameter('est') == 0) {
            $est="P";
        }
        else if($this->getRequestParameter('est') == 1) {
            $est="E";
        }
        else {
            $est="T";
        }
```

Figura # 14 Actualizar planilla acciones correctivas-a.

```
        $acciones = Doctrine::getTable('AccionesCorrectivas')
                ->createQuery('a')
                ->where('a.num = ?', $num)
                ->execute();
        $acciones[0]->redactor = $red;
        $acciones[0]->cargo = $cargo;
        $acciones[0]->num = $num_ac;
        $acciones[0]->accion = $acc;
        $acciones[0]->elemento = $elem;
        $acciones[0]->fecha = $fecha;
        $acciones[0]->importancia = $imp;
        $acciones[0]->estado = $est;
        $acciones[0]->save();
        $this->getUser()->setFlash('notice', sprintf('La acción correctiva "'.
        $this->getRequestParameter('num').'" fue actualizada correctamente'));
        $this->redirect('auditoria_rev/reg_plan_rev');
    }
    else
    {
        $this->acciones = Doctrine::getTable('AccionesCorrectivas')
            ->createQuery('c')
            ->where('c.num_control = ?', $this->getRequestParameter('nume'))
            ->execute();
        $this->recurso = Doctrine::getTable('RecursosHumanos')
            ->createQuery('a')
            ->execute();
        $this->proyecto = Doctrine::getTable('ProyectosExisten')
            ->createQuery('a')
            ->execute();
    }
}
```

Figura # 15 Actualizar planilla acciones correctivas-b.