UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9

TÍTULO: Implementación de los módulos IEA del Grupo de Calidad de la Facultad 9.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA

AUTOR: Felix Noel Abelardo Santana.

TUTOR: Ing. Yoandry Lazo Nodarse.

COTUTOR: Ing. Yudiel Rodríguez Larrazabal.

Ciudad de la Habana junio de 2010

Declaración de autoría

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

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

_____ _____

Firma del Autor Firma del Autor

_____ _____

Firma del Tutor Firma del Cotutor

Frase

"... Por lo tanto la lucha por la calidad del producto es una lucha revolucionaria y de vanguardia"

Ernesto "Che" Guevara

Agradecimientos

A mis padres, por su dedicación y apoyo constante.

A mis amistades por estar siempre ahí cuando las necesitamos…

A toda mi familia por su apoyo en especial a mi abuela y a mi abuelo por su preocupación y apoyo constante…

A todas las personas que han contribuido de alguna manera en mi formación profesional a través de sus enseñanzas y lecciones, a todas gracias por guiarme siempre por el buen camino…

A la Revolución, a Fidel y a Raúl por hacer posible ésta Universidad de excelencia…

Dedicatoria

A mis padres: Por haberme dado todo lo que soy como persona, por haberme inculcado los principios, los valores, la perseverancia y el empeño, acompañado siempre de una gran dosis de amor, sin pedir nunca nada a cambio.

A mi hermano: Gran persona que me ha hecho seguir adelante en los momentos más difíciles con su gran dosis de amor jaranearía.

A mis familiares: A mis tíos y tías por su gran a apoyo, a mi prima querida Rosanny y a mis queridos abuelos Teresa Sánchez López y Nicolás Santana Navarro.

A mi novia: Yaimit, por su paciencia, por su comprensión, por su empeño, por su amor, dedicación y por ser tal y como es.

A mis amigos: Que siempre estuvieron ahí cuando los necesitaba, dando todo de ellos incondicionalmente.

A los profesores: A los profesores que de una forma u otra tuvieron que ver con mi educación en estos 5 años de universidad.

Resumen

Hoy en día, con el creciente auge de la informatización en todas las esferas de la sociedad y el acelerado desarrollo de las Tecnologías de la Informática y las Comunicaciones, la mayoría de las empresas e instituciones necesitan automatizar los procesos de trabajo que realizan. Debido al creciente y constante volumen de documentación que se tramita en el Grupo de Calidad perteneciente a la Facultad 9, en la Universidad de las Ciencias Informáticas (UCI), se hace necesario automatizar gran parte de los procesos allí realizados. Por tanto, el objetivo principal del presente trabajo se centra en la implementación de los módulos Procedimientos de Inicio de Proyectos y Procedimiento de Estrategia de Prueba del portal de Calidad de la Facultad 9, en la cual estas actividades se desarrollan de forma lenta y compleja.

El objetivo general de esta investigación es diseñar el portal de Calidad de la Facultad 9 y ejecutar la primera iteración de la fase de construcción, logrando una correspondencia con los requerimientos de los usuarios del mismo. Para lograr las características anteriores de la forma más eficiente y novedosa se decide hacer uso del Framework Symfony para el desarrollo del portal, el cual provee una serie de funcionalidades como el manejo de contenido, seguridad basada en credenciales y escape de los datos introducidos en los formularios.

El documento recoge un estudio sobre otros sistemas de gestión de la Calidad en el ámbito nacional, internacional y en la UCI. En el mismo quedan plasmadas las características de las herramientas usadas para la solución dada, un estudio de las técnicas, lenguajes de programación y del IDE usado. Además en documentos anexos a este se recoge la validación de la solución mediante los casos de prueba de caja negra.

El resultado obtenido fue un portal completamente actualizable mediante un sistema de administración basado en usuarios y credenciales, logrando automatizar la gestión de los procedimientos antes mencionados que se realizaban en el Grupo de Calidad de la Facultad 9.

PALABRAS CLAVE:

Calidad, Procedimientos de Inicio de Proyectos, Procedimiento de Estrategia de Prueba**.**

Índice

Índice de tablas

Introducción

A lo largo de toda la historia el hombre ha buscado la perfección de su trabajo, por lo que ha crecido el interés por lo que realiza y la necesidad de asumir responsabilidades sobre la labor realizada, esto se fue derivando poco a poco en la obtención de la mejoría de sus acciones. En el camino emprendido por la humanidad para alcanzar la calidad ha cursado por varios estadíos comenzando por las inspecciones y terminando en la calidad total, pasando por el control estadístico de la calidad, los sistemas de aseguramiento de la calidad y los sistemas de gestión de la calidad.

Para un mejor entendimiento de lo antes expresado se darán a conocer varios conceptos:

Calidad: ...la calidad se refiere a las características mensurables -cosas que se pueden comparar con estándares conocidos como longitud, color, propiedades eléctricas, maleabilidad, etc. (S: Pressman)

Calidad total: Conjunto de condiciones que permiten asegurar la mejora continua de los procedimientos, procesos, actividades y manejo de recursos públicos por las dependencias y entidades del sector público presupuestario, con la finalidad de controlar, prevenir y eliminar cualquier tipo de deficiencia en la presentación o producción de los bienes y servicios que dan a sus clientes o usuarios, con el propósito de proporcionar la máxima satisfacción con la mayor eficacia y eficiencia. (definición, 2010)

Control estadístico de la calidad: Un enfoque estadístico que determina cuantos artículos de un lote inspeccionado, con seguridad cumplen con los estándares de calidad del producto. (Pirámide Digital, 2007)

Aseguramiento de la calidad: Conjunto de acciones planificadas y sistemáticas que son necesarias para proporcionar la adecuada confianza de que un producto o servicio satisfará los requisitos dados sobre la calidad. (Diccionario de términos de certificación, 2010)

Sistema de gestión de calidad: Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad. (ISO, 2000)

A partir de la Resolución No. 63 del 2005 del Ministerio de la Informáticas y las Comunicaciones de Cuba fue creado el Centro Nacional de Calidad de Software (Calisoft), compuesta por un grupo nacional de expertos en el tema, que juegan el rol de “tercero” confiable para la realización de las pruebas de liberación de todos los entregables y producto final, las pruebas de aceptación con el cliente y pruebas piloto con el usuario final. La misión de Calisoft es ser líder en la Gestión del Conocimiento en el área del aseguramiento de la calidad que pueda ser utilizado para la mejora de las organizaciones y la excelencia empresarial en la Industria de Software logrando mejoras competitivas medibles.

Pero el trabajo de Calisoft es a nivel de país y de universidad por lo que en la UCI en cada facultad existe un Grupo de Calidad. El presente trabajo está enmarcado en el Grupo de Calidad de la Facultad 9, el cual al igual que la universidad está inmerso en la producción de software y trata de que su producto sea cada vez mejor y con mayor calidad y aceptación por parte de los usuarios o clientes que usarán el mismo, por lo que la calidad es un elemento fundamental en el desarrollo de estos.

El Grupo de Calidad de la Facultad 9 al igual que sus homólogos es el encargado de velar por la calidad de los proyectos en esa facultad, en dicho proyecto se llevan a cabo varias actividades, dentro de las cuales tenemos el Procedimiento de Inicio de Proyecto del que se derivan varios artefactos relacionados con la certificación de los proyectos productivos; por otra parte tenemos el Procedimiento de Estrategia de Prueba en el que se generan artefactos relacionados con las pruebas que se le realizan a los proyectos productivos.

El Grupo de Calidad de la Facultad 9 argumenta que la documentación no es accesible para todas las personas que la necesiten, dificultándose la gestión de la misma y provocando lo siguiente:

- ✓ Demora de la gestión de los procedimientos.
- ✓ Lentitud en la búsqueda y recuperación de información.
- ✓ Necesidad de consultar los archivos para ver una información determinada.
- ✓ Duplicado de información almacenada.

Teniendo en cuenta lo mencionado anteriormente se define como **problema a resolver** el difícil acceso a la documentación y el retraso en tiempo con la gestión de la información referente al Procedimiento de Inicio de Proyecto, y a la Estrategia de Prueba en el Grupo de Calidad de la Facultad 9 (Calidad9), lo que impide que se lleven a cabo con mayor rapidez estas actividades.

Teniendo como **objetivo general** de la investigación implementar los módulos Inicio de Proyecto y Estrategia de Prueba (IEA) en el Grupo de Calidad de la Facultad 9, para lograr que la gestión de la información se lleve a cabo de una manera más rápida.

Para darle solución a este problema se define como **objeto de estudio** el Proceso de gestión de información en el Grupo de Calidad de la Facultad 9.

El resultado de la investigación tendrá como **campo de acción** la automatización de los Procedimientos de Inicio de Proyecto y Estrategia de Prueba en el Grupo de Calidad de la Facultad 9.

Dando lugar a definir como **idea a defender** que si se realiza un software que permita la gestión de la información de la Estrategia de Prueba y del Procedimiento de Inicio de Proyecto en la Facultad 9, permitirá agilizar los procesos de gestión en cada una de estas áreas.

Durante el desarrollo de esta investigación, y para dar respuesta al problema científico planteado, se aplicarán los siguientes **métodos científicos**:

- ✓ **Métodos teóricos:**

  - Histórico-Lógico: Mediante el cual se analizará la trayectoria y desarrollo del problema y como se pone de manifiesto la lógica interna de su desarrollo en la Facultad 9.

  - Analítico-Sintético: Con su utilización se podrá estudiar el problema con mayor profundidad debido a la división mental y posterior unión de las partes analizadas previamente, para dar la solución adecuada al problema científico. Este método facilita el entendimiento del fenómeno en el que se trabaja debido a la división del mismo que se realiza, para de esta forma descubrir sus principales características así como las generales.

- ✓ **Métodos empíricos:**

  - Encuesta: Con la realización de encuestas directas, elaborando preguntas cerradas, se permitirá estandarizar los datos recogidos para un estudio posterior de los mismos y de esta manera comprobar o no el cumplimiento del objetivo general. Para llevar a cabo la encuesta se seleccionó una población integrada por los líderes de todas las áreas que integran el Grupo de Calidad de la

Facultad 9, tomando como muestra la totalidad de ellos permitiendo así que los resultados se acerquen más a la realidad. Para la realización de la encuesta se conformó la misma con preguntas que son respondidas marcando una opción de las brindadas por esta, para evitar que el encuestado se aleje de la respuesta que se espera que este de.

Como posibles resultados de la investigación se tendrá:

1. Diagrama de Componentes.
2. Código Fuente.
3. Diseños de casos de pruebas.
4. Prototipo funcional del módulo IEA.

Las **tareas de la investigación** que se proponen para lograr la realización de la misma son:

1. Actualizar el conocimiento en torno al proceso de gestión de la información referente al módulo IEA.
2. Determinar el tipo de aplicación a implementar.
3. Determinar las herramientas para el desarrollo de la aplicación.
4. Determinar el lenguaje de programación para el desarrollo de aplicación.
5. Implementar el módulo IEA.

Este trabajo investigativo tiene como antecedentes: El portal del Centro Nacional de Calidad de Software (Calisoft), el portal del Grupo de Calidad de la Facultad 7 y el Portal de Calidad de la Facultad 1; también tiene como antecedentes al portal Calidad del Software con la dirección www.calidaddelsoftware.com y el Portal de Calidad del Software para Latinoamérica con la dirección www.esipcla.com, estos dos últimos pertenecen a la esfera internacional.

Capítulo 1: Fundamentación Teórica.

1.1. Introducción

Este capítulo aborda los temas referentes al estado del arte del tema tratado, las tendencias actuales que existen sobre las aplicaciones de gestión de información de calidad, a nivel nacional, internacional y en la UCI; así como las ventajas y desventajas de los diferentes tipos de aplicaciones y el uso de herramientas para el desarrollo de las mismas. Además se describe la justificación del uso de las herramientas, lenguaje de programación y estándares de codificación en el desarrollo de la aplicación.

1.2. Conceptos asociados al dominio del problema

Antes de entrar en los detalles propios de la aplicación a desarrollar, cabe destacar, que para cumplir los objetivos de este trabajo, se debe desarrollar un sistema que les posibilite a múltiples usuarios tener acceso al mismo tiempo a la aplicación, brindando la obtención rápida de la información, sin necesidad de compartir escritorios, ni del intercambio físico directo entre los aseguradores de calidad de cada proyecto en la facultad y sus usuarios, por lo que se emprende la tarea de buscar y demostrar que tipo de aplicación es la más idónea para satisfacer estos objetivos.

1.2.1. Aplicación

Una aplicación es un programa de computadora que se utiliza como herramienta para una operación o tarea específica. Para la informática, una aplicación es uno de diversos tipos de programas de computación diseñados especialmente para cumplimentar una función o actuar como herramienta para acciones puntuales del usuario. (Victoria, 2009)

Aplicación es cualquier programa informático que se ejecute en un sistema operativo y que haga una función específica para un usuario. Suele resultar una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, o la gestión de información. Algunos ejemplos de los programas de aplicación más comunes son los procesadores de textos, hojas de cálculo, y base de datos. En función de las tareas que desempeñe la aplicación, será más general o más específica. Cuanto más general es, más casos puede abarcar, pero a veces es más interesante una aplicación específica, ya que está desarrollada casi a la medida de las necesidades y suele ofrecer más soluciones para un mismo problema. Las grandes compañías comerciales desarrollan

paquetes de aplicaciones, es decir, agrupan una serie de programas de distinta naturaleza para satisfacer al mayor número de usuarios.

1.2.2. Aplicación de escritorio

Una aplicación de escritorio es toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea WINDOWS, LINUX, MAC ó en todas ellas. El desarrollo sobre una plataforma, normalmente, implica que la aplicación no pueda ser ejecutada en otras. Una aplicación de escritorio posee varias características dentro de las cuales tenemos que requiere de su instalación en el disco duro de la computadora, son independientes de internet por lo que su ejecución no depende de si se posee conexión o no en determinado momento, poseen gran velocidad de acceso aunque esto es relativo según el sistema operativo utilizado y la potencia del procesador de la máquina, con una computadora de potencia media la aplicación debe ejecutarse de manera óptima, debe disponer de gran cantidad de opciones que acompañan el objetivo de la herramienta dentro de los cuales podemos tener las opciones de: creación, edición y almacenamiento de información.

La computadora personal en la actualidad ha tenido un enorme impacto por el solo hecho de haber puesto al servicio de un usuario normal el poder de la misma en vez de una terminal simple, esto ha potenciado una gran diversidad de usos dentro de los que encontramos la realización de aplicaciones de escritorio para el uso de otras personas, empresas y todos los necesitados de las mismas.

1.2.3. Aplicación web

Se denominan aplicaciones web a aquellos sistemas de software que los usuarios pueden utilizar accediendo a un servidor web mediante un navegador de Internet. Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Es decir, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a que el navegador web como cliente ligero es muy práctico y no hay que instalar ningún programa con CD, descargar o actualizar el software para su utilización. La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en los potenciales clientes es otra razón por la cual se utilizan.

1.3. Aplicación de escritorio vs. Aplicación web.

A la hora de seleccionar el tipo de aplicación que más se ajusta a las necesidades del Grupo de Calidad de la Facultad 9 se tuvieron en cuenta varios factores que influyen en el buen desempeño que se espera del producto final de la presente investigación, por lo que se realizó una comparación donde se tienen factores como parámetros. La comparación realizada se encuentra plasmada en la tabla que se presenta a continuación:

	Sistema operativo	Actualizaciones	Acceso a la aplicación	Control de acceso	Datos duplicados
Aplicaciones de Escritorio	Existencia de una versión para cada plataforma.	Traumas a la hora de realizar actualizaciones al programa, las instalaciones están diseminadas.	Se limitan a manejar datos en el sistema de archivos local.	La administración de la seguridad es difícil.	La información se encuentra toda en cada máquina que use la aplicación.
Aplicaciones Web	Aplicaciones multiplataforma.	Sencillas, sin necesidad de descargas.	Desde cualquier lugar.	Facilidad de control de acceso.	La información esta toda en un mismo servidor.

Tabla 1: Aplicación de escritorio vs. Aplicación web

Las aplicaciones de escritorio se han usado y se seguirán usando y tienen un campo enorme, no todo está en la Web, hay cosas que se necesita que se ejecuten estrictamente en una máquina en específico para aprovechar el poder que se tiene al alcance. Pero la fusión e integración de servicios de las computadoras, las aplicaciones de escritorio y la extensión de las facultades de comunicación con las aplicaciones Web que hace posible la Internet, es la plataforma óptima que sirve de infraestructura para todos los usuarios desde el tipo institucional o personal. La comparación realizada arrojó como resultado que la aplicación más idónea a implementar es una aplicación web, debido a las grandes potencialidades que poseen las mismas a la hora de realizar actualizaciones y de controlar el acceso a la aplicación así

como su usabilidad en los distintos sistemas operativos, por lo que este tipo de aplicación es el que más se ajusta a las necesidades que se tienen en el Grupo de Calidad de la Facultad 9.

1.4. Portal

Un portal se puede definir como la evolución del concepto de "Web Site", en donde el Web se ha convertido en el punto de entrada a un conjunto de servicios e información, a los que se accede de formas sencilla, unificada y segura. (Carrion, y otros, 2000)

Sitio web que sirve de punto de partida para navegar por Internet. Los portales ofrecen una gran diversidad de servicios: listado de sitios web, buscador, noticias, e-mail, información meteorológica, chat, newsgroups (grupos de discusión) y comercio electrónico. En muchos casos el usuario puede personalizar la presentación del portal. Algunos de los más conocidos son AltaVista, Yahoo!, Netscape y Microsoft. (Informático, 2009)

El nombre portal tiene que ver con la idea de que es una puerta grande a múltiples servicios y oportunidades para el usuario, quien lo usa como referencia para navegar, descubrir nuevas posibilidades y realizar búsquedas en la web u obtener información de esta. Un portal puede tener el objetivo de resolver un tipo de necesidad específica de un grupo de personas o comunidad con intereses en particular, pero también con frecuencia los portales son usados por marcas o empresas para ofrecer una experiencia participativa al usuario en general. Su nacimiento está enmarcado a mediados de la década de los 90, cuando los buscadores liderados por Yahoo y en menos medida por AltaVista, comenzaron a ampliar sus páginas principales incluyendo principalmente índices y directorios ofreciendo algunos contenidos de reconocido interés para sus visitantes.

Existen varias clasificaciones de portales de las que se abordarán 1.

1. En cuanto a su clasificación los portales pueden ser:

   - ✓ Horizontales: Tienen un propósito general y una audiencia amplia por lo que son más masivos, tratando de llegar a toda la gente con muchas cosas, teniendo como ejemplo el portal de Yahoo y AltaVista.

- ✓ Verticales: Se dirigen a usuarios para ofrecer contenido dentro de un tema específico como puede ser un portal de música, empleo, inmobiliario, un portal de finanzas personales, arte o de deportes.
  - ➢ Portal Intranet: Comunicación corporativa para los empleados.
  - ➢ Portal Extranet: Comunicación corporativa para los proveedores.
  - ➢ Portal Vertical: Comunicación corporativa con clientes.
- ✓ Diagonales: Son una mezcla entre el portal horizontal y el vertical. Se trata de portales que utilizan redes sociales o aplicaciones generalistas como Facebook.

Todas estas clasificaciones pueden ser incluso complementarias. El tipo de portal a implementado es un portal vertical, ya que se analizó, diseñó e implementó el mismo para un tema en específico que es la Calidad de Software, se escogió esta clasificación de portal porque es el que más se ajusta para darle solución a las necesidades que se tienen en el Grupo de Calidad de la Facultad 9.

Características de los portales:

- ✓ Un solo punto de acceso a todos los contenidos que pertenecen al dominio del portal, siendo la administración de contenidos una parte muy importante dentro del portal.
- ✓ Interacción personalizada con los servicios que ofrece el portal.
- ✓ Acceso a información de fuentes diversas, agregada y categorizada.
- ✓ Integración de herramientas de colaboración como grupos de trabajo, comunidades, foros o grupos de discusión y chat's.
- ✓ Integración con Aplicaciones y Sistemas de workflow (flujo de trabajo).

1.5. Portal Web

Un Portal Web es un sitio web que actúa como un único punto de acceso para una gran variedad de información. (De Seta, 2009)

Un portal web es un sitio en la web que proporciona conexión, información, participación y múltiples servicios a los usuarios, como la búsqueda de información, acceder a diversos servicios, recursos o aplicaciones desde un mismo lugar, teniendo como objetivo incrementar la intensidad del uso del mismo mediante la diversificación de servicios. Se le llama portal web al tipo de sitios web que tienen el propósito

de centralizar un conjunto de servicios y recursos de manera integrada para el usuario, a menudo posibilitando que el mismo se informe, participe, opine o acceda a múltiples aplicaciones.

El portal web surge ante el problema de brindar a un grupo de usuarios acceso a una diversa cantidad de recursos y servicios informáticos de forma integrada y sencilla. En él se suelen encontrar herramientas para compra electrónica, programas, documentos de toda clase, foros de usuarios, y buscadores. Las instituciones, ya sean privadas o públicas, suelen necesitar crear Portales Web para dar acceso a sus contenidos informáticos, dado que una página muy sencilla en general no sería suficiente.

1.5.1. Portal de gestión de información

Gestión de la información: Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada. (Woodman, 1985)

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. En el centro de la gestión de la información se encuentra la gestión de la documentación (la información que queda plasmada en documentos) y que pude ser de tres tipos: interna, externa y pública.

La calidad es un problema de orientación, de liderazgo, de participación de todos los implicados en la misma. En cualquier caso, la mejora de la calidad es un proceso sin fin, que debe llevarse paso a paso y del que no se pueden esperar resultados inmediatos. En el mundo actual, la gestión del conocimiento adquiere nuevas características determinadas por la gestión de la información y de la calidad. En las organizaciones más modernas cohabitan, indisolublemente ligadas, la gestión de información, del conocimiento y de la calidad. Para darle solución a la problemática planteada en este trabajo se realizó un portal web de gestión de información, que como el nombre lo dice en el mismo se puede obtener información sobre calidad así como planillas para la realización de tareas relacionadas con sus procedimientos.

1.6. Análisis de soluciones existentes

En la actualidad existen una gran cantidad de portales web que gestionan información referente a la calidad de un software dentro de los cuales tenemos:

1.6.1. En el ámbito internacional

Calidaddelsoftware.com: (español).Su principal función es la publicación de libros, documentos y artículos referentes a estándares, métricas, normas y la organización de eventos y cursos. También se muestran definiciones y conceptos necesarios para entender el contenido del portal. Se publican modelos a seguir para obtener productos con calidad, realizan consultas y foros donde se puede debatir cualquier duda que tenga el usuario.

Esipcla.com: El portal representa una división para Latinoamérica responsable de representar la European Software Institute (ESI) en la región, colaborando con las empresas y gobiernos en diferentes campos de acción definidos por la ESI. Su principal actividad se basa en ayudar a la industria del software en sus objetivos de producir mejor software de mayor calidad, en el tiempo previsto, y con un menor costo.

1.6.2. En el ámbito nacional y enmarcado en la UCI

Calisoft.uci.cu: Su función es la de garantizar el crecimiento continuo de una producción de software con calidad en la Universidad de las Ciencias Informáticas; a través de la definición de procesos siguiendo las especificaciones de metodologías, estándares y modelos de desarrollo de software, brindando asesorías, entrenamiento, métodos de medición y servicios de verificación-validación a las diferentes entidades.

Portal de Calidad Facultad 2: Sus funciones principales se enmarcan en la consulta online de documentación actualizada sobre el tema de la Calidad de Software, publicación de información y noticias sobre trabajo que se desarrolla y temas de interés para el proyecto, plantillas y documentación sobre trabajos anteriores, así como record de los proyectos revisados, gestión de la ubicación de puestos de trabajo del proyecto.

Portal de Calidad Facultad 7: Sus funciones principales se enmarcan en la realización de examen de admisión, gestión de pruebas de admisión, cursos de formación, horario de producción, solicitudes de pruebas a un producto y las promociones y convocatorias de eventos, así como las consultas a los cursos

asignados. También poseen documentación actualizada sobre el tema de la Calidad de Software y temas de interés para el proyecto, plantillas y documentación sobre trabajos anteriores.

El Grupo de Calidad de la Facultad 9 posee características similares a las resueltas por estos portales pero no son todas las que se puede satisfacer por lo que se emprendió la tarea de crear su propio portal de gestión de información que reúna las particularidades que realmente requiere la facultad para llevar a cabo un buen proceso de gestión de la información.

1.7. Tecnologías y tendencias actuales a considerar

Para la realización de una aplicación hay que tener en cuenta por donde se mueve el mundo en torno a las herramientas a utilizar para el desarrollo de la misma para que de esta forma el trabajo se haga más asequible y con mayor calidad. Se deben utilizar tecnologías avanzadas y en continuo progreso para que de esta forma la aplicación cumpla con las expectativas de los usuarios finales y además se pueda actualizar la misma con gran facilidad. Para la realización de la aplicación web de gestión de información se analizaron algunos de los lenguajes de programación, Framework, IDE y servidores web existentes en aras de determinar cuál o cuáles son los más idóneos para la realización de la tarea a emprender.

1.8. Fundamentación de las tecnologías y herramientas propuestas a utilizar

1.8.1. Lenguajes o tecnologías de Programación

1.8.1.1. PHP 5.3 (Personal Home Page)

Es un lenguaje de programación libre para la creación de páginas webs dinámicas que permite la creación de aplicaciones con interfaz gráfica, conexión a servidores de base de datos como Oracle, MySQL, Postgres y puede ser ejecutado en sistemas Windows, Linux y Mac OS. Es un lenguaje de código abierto sumamente popular especialmente utilizado para desarrollar aplicaciones que se ejecutan en servidores Web y puede ser integrado en HTML. Ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación. Su elegante diseño lo hace perceptiblemente más fácil de mantener y ponerse al día en comparación con el código de otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que

los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje " OpenSource " interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. (Achour, y otros, 2007)

La característica más importante de PHP es que permite combinar código HTML y código PHP en una misma página web. Otra característica interesante de PHP es que permite realizar inclusiones dentro de un script PHP, de esta forma una página se puede dividir en varios módulos PHP que se desarrollan en forma independiente, además pueden desarrollarse componentes reusables en otras páginas o incluso en otros sitios.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aún estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes mediante la utilización de Framework dentro de los cuales se destaca el Symfony. En cuanto a la seguridad PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI (Interfaz de entrada común o Common Gateway Interface) separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor Web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

1.8.1.2. ASP (Active Server Pages o Active Server Pages)

ASP (Servidor de Páginas Activas) es el nombre de una tecnología al lado del servidor para crear páginas web generadas dinámicamente y componentes de servidor ActiveX reutilizables, así como páginas interactivas en la web, desarrollada por Microsoft para ser usada en páginas alojadas en un servidor con sistema operativo de Microsoft. La potencia del ASP es asombrosa, con él pueden llegar a construirse verdaderas aplicaciones cliente-servidor, posee gran facilidad de uso y prestaciones. Cuando un usuario solicita un archivo .asp en su navegador, el servidor interpreta los comandos y genera la página que le enviará finalmente. Con este lenguaje, que permite además utilizar códigos de Visual Basic, JavaScript y otros lenguajes, se crean al igual que con otros, lo que se ha llamado sitios dinámicos.

ASP no es realmente un lenguaje como tal, es el acrónimo de Active Server Pages, el lenguaje usado en realidad para programar ASP es Visual Basic Script o JScript. El mayor inconveniente de ASP es que se trata de un sistema propietario que es usado nativamente sólo por Microsoft Internet Information Server (IIS). Esto limita su disponibilidad a servidores basados en Win32. Existe un par de proyectos en desarrollo que permiten que ASP corra en otros entornos y servidores web...Se dice que ASP es un lenguaje más lento y pesado que PHP, y también menos estable. Algunas de las ventajas de ASP consisten en que debido a que usa principalmente VBScript, es relativamente simple tratar con el lenguaje si usted ya conoce cómo programar en Visual Basic. El soporte de ASP también se encuentra habilitado por defecto en el servidor IIS, facilitando su instalación y ejecución. Los componentes integrados en ASP son bastante limitados, de modo que si necesita usar características "avanzadas", como interactuar con servidores FTP, necesita comprar componentes adicionales.(Achour, y otros, 2007)

1.8.1.3. JSP (Páginas de Servidor Java)

Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems. JSP permite la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

JSP puede considerarse como una manera alternativa, y simplificada, de construir servlets. Es por ello que una página JSP puede hacer todo lo que un servlet puede hacer, y viceversa. Cada versión de la

especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de servlets. El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

1.8.2. Comparación para seleccionar la tecnología o lenguaje de programación a utilizar.

Las tecnologías o lenguajes de programación que más se ajustan a las necesidades requeridas son: PHP, ASP y JSP, por lo que a continuación se expondrán las ventajas y desventajas de la utilización da cada uno de ellos para la realización del portal web. En la tabla que se presenta a continuación se realiza una comparación entre las dos tecnologías o lenguajes de programación para de esta forma seleccionar el más idóneo.

	Sistema Operativo	**Tipo de software**	**Base de Datos**	**Documentación**	**Curva de aprendizaje**
PHP	Es un lenguaje multiplataforma.	Software libre.	Interfaces para una gran cantidad de sistemas de Base de Datos diferentes.	Amplia documentación en su página oficial y difundida por toda la web.	Sencillo de aprender.
ASP	Windows.	Software propietario.	Microsoft SQL Server.	Amplia.	Sencillo de aprender.
JSP	Es un lenguaje multiplataforma.	Software libre.	Interfaces para una gran cantidad de sistemas de Base de Datos diferentes.	Amplia.	Complejo, exige el conocimiento del lenguaje Java.

Tabla 2: Comparación para seleccionar la tecnología o lenguaje de programación a utilizar

Como resultado de la comparación realizada anteriormente entre PHP, ASP y JSP, se seleccionó el PHP 5.3 como el más idóneo para la programación del portal web debido a las características que posee dentro de las cuales podemos mencionar que permite la conexión a servidores de base de datos como Postgres y puede ser ejecutado en sistemas Windows, Linux y Mac OS, es un lenguaje de código abierto especialmente utilizado para desarrollar aplicaciones que se ejecutan en servidores web, por el hecho de ser libre se presenta como una alternativa de fácil acceso para todos los desarrolladores; ofrece una solución simple y universal para la creación de páginas dinámicas en la web. Mediante PHP pueden desarrollarse componentes reusables en otras páginas o incluso en otros sitios, este lenguaje de programación permite aplicar técnicas de programación orientada a objetos manteniendo un bajo consumo de recursos de la máquina teniendo gran seguridad y poca probabilidad de corromper los datos. Mientras que de ASP se puede expresar que su mayor inconveniente es que se trata de un sistema propietario que es usado nativamente sólo por servidores web con sistema operativo de Microsoft y que es un lenguaje más lento, pesado y menos estable que PHP.

1.8.3. Framework

La palabra inglesa Framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software un Framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos de software concretos, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Un Framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

1.8.3.1. Prado (Desarrollo rápido de aplicaciones orientado a objetos con PHP)

PRADO es un Framework para PHP basado en componentes y en eventos. La primera versión se realizó para PHP4, pero se reescribió completamente para PHP5. Entre las características se encuentra la separación entre la presentación y la lógica, su arquitectura modular configurable, componentes web, internalización y localización, manejo de errores, logs y caché.

Entre los objetos que esta plataforma de desarrollo ofrece, se encuentra el acceso a bases de datos, formularios y controles web. Además Prado soporta componentes AJAX, permitiendo la personalización en la localización de errores y excepciones.

Este Framework posee como desventaja que se enfoca en lo que es la vista del proyecto y no en la parte de acceso a los datos que se presentan en la aplicación, por lo que si se quiere desarrollar un software que se enfoque en los datos que se guardan no se recomienda su utilización.

Prado está basado en componentes eventos con el objetivo de acelerar el desarrollo de aplicaciones web usando PHP5. El concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros. Este Framework combina especificaciones en un archivo XML, plantillas HTML y una clase PHP. Prado, cuenta con soporte para AJAX, validación, autentificación, plantillas, múltiples bases de datos.

1.8.3.2. Zend Framework

Zend Framework es un Framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Es una implementación que usa código orientado a objetos. La estructura de sus componentes es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible Framework de aplicaciones web al combinarse. Zend Framework ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y StrikeIron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

1.8.3.3. Symfony

Un Framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un Framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un Framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. Symfony es un completo Framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix y Linux, como en plataformas Windows. A continuación se muestran algunas de sus características:

Symfony está diseñado para que se ajuste a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas.
- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

Debido a las características expresadas con anterioridad sobre Symfony así como las potencialidades que presenta, se escoge el mismo para la realización del portal web.

1.8.4. IDE (Entorno de Desarrollo Integrado)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

1.8.4.1. Zend Studio

Zend Studio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación PHP. Debido a que está creado completamente en Java permite que existan versiones del mismo tanto para Windows, como para Linux y Mac OS. Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata de un software comercial, lo cual contrasta con el hecho de que PHP es software libre.

Este software permite hacer depuraciones simples de scripts desde la parte del cliente, para la realización de una depuración más potente se debe disponer de la parte del servidor, que instala Apache u otro servidor WEB, y el módulo PHP. 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, esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que se vayan creando, incluso en páginas que se tengan incluidas con la función include(), además posee resaltado de sintaxis, auto completamiento de código, emparejamiento de paréntesis y corchetes, detección de errores de sintaxis en tiempo real, optimizadores de código y herramientas de base de datos. Zend Studio fue diseñado para usarse con el lenguaje PHP, sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Java script y XML. Esta herramienta permite agilizar el desarrollo web y simplificar proyectos complejos, posee integración con programas

controladores de versiones como el subversión y con servicios de FTP, además posee soporte para Web Services, PHP4, PHP5 y SQL.

Características:

- ✓ Compatible con las plataformas Linux, MAC y Windows.
- ✓ Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos.
- ✓ Permite agilizar el desarrollo de una web y simplificar proyectos complejos.
- ✓ Excelente completamiento de código.
- ✓ Coloreado en la sintaxis del código.
- ✓ Incorpora el manual de PHP.

1.8.4.2. Aptana

Aptana es un IDE para la programación de aplicaciones web dinámicas, con soporte especial para Ajax/Java Script, es uno de los mejores programas que utiliza AJAX. Es un entorno de desarrollo basado en Eclipse que soporta HTML, DOM, Java Script y CSS, y que además posee correctores para estos lenguajes. Ilustra diferentes 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. Aptana tiene como ventaja que puede extender sus funcionalidades mediante plugins, lo que hace que el trabajo sea más cómodo en algunos lenguajes de programación como PHP, Python o Ruby. Cuenta con una edición gratis para la comunidad y otra profesional de pago.

1.8.5. Zend Studio vs. Aptana

Las dos IDE que más se ajustan a las necesidades requeridas son: Zend Studio y Aptana, por lo que a continuación se realiza una comparación entre los dos para de esta forma seleccionar el más idóneo.

	Sistema Operativo	**Tipo de software**	**Consumo de recursos**	**Presencia de depurador**	**Completamiento de código**
Zend Studio	Multiplataforma.	Software comercial.	Poco consumo de recursos.	Cuenta con un buen depurador.	Posee auto completamiento de código.

Aptana	Multiplataforma.	Software libre.	Gran consumo de recursos.	Cuenta con depurador.	Posee auto completamiento de código.

Tabla 3: Zend Studio vs. Aptana

Como resultado de la comparación realizada anteriormente entre los dos IDE de programación se seleccionó el Zend Studio como el más idóneo para la programación del portal web. Debido a las características que posee dentro de las cuales podemos mencionar el bajo consumo de recurso de la máquina, que es multiplataforma y que con anterioridad se seleccionó como lenguaje de programación el PHP, y el Zend Studio se creó para el desarrollo de aplicaciones Web utilizando PHP, contando con completamiento de código no solo de las variables locales sino también de las funciones definidas en otros ficheros relacionados con el trabajo que se está realizando. Además Zend Studio contiene una ayuda contextual con todas las librerías de funciones del lenguaje PHP que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir.

1.8.6. Servidores Web

1.8.6.1. AOLserver (Servidor web de América Latina Online)

AOLserver es el servidor web de código abierto de América Online. Tiene procesamiento multihilo, y se usa para sitios web dinámicos de gran tamaño. Se distribuye bajo la licencia AOLserver Public License, que es similar a la de Mozilla (Mozilla Public License o Licencia Pública de Mozilla). Este servidor web se desarrolló originalmente por NaviSoft con el nombre "NaviServer", pero este cambió cuando AOL compró la compañía en 1995. Philip Greenspun convenció a America Online para que liberalizase el código del programa en 1999. AOLserver fue el primer servidor HTTP en combinar el procesamiento multihilo, con un lenguaje interpretado de serie, y el procesamiento de colas de conexiones persistentes a base de datos. Para los sitios web con bases de datos, esto permite mejorar el rendimiento hasta cien veces más que la práctica habitual, las cuales abrían una nueva conexión a la base de datos en cada petición de página. Hay otros servidores HTTP que consiguen un rendimiento similar con una arquitectura similar, pero AOLserver está varios años por delante de la competencia.

1.8.6.2. Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix, Windows y Macintosh. El nombre Apache es un acrónimo de (A Patchy Server) servidor de remiendos, es decir un servidor construido con código preexistente y piezas y parches de código. Es el ejemplo de software libre de mayor éxito, por delante incluso del Kernel Linux. Desde hace años, más del 60% de los servidores web de Internet emplean Apache, servidor originalmente pensado para el entorno Linux y que dispone de versiones para Windows.Apache es usado para muchas tareas donde el contenido de una página web necesita ser puesto a disposición de muchas personas en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia una red. La arquitectura del servidor Apache es muy modular.

Una vez instalado Apache, se dispone de un servidor web y otro de base de datos, configurado de manera local, que permite realizar todas las pruebas necesarias en las web antes de lanzarla a la red. Debido a las características, expresadas con anterioridad sobre Apache, así como las potencialidades que presenta se selecciona el mismo para la realización del portal web.

1.8.7. Gestor de Base de Datos

1.8.7.1. PostgreSQL

El SGBD relacional orientado a objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley, distribuida bajo licencia BSD. Con más de una década de desarrollo tras él, PostgreSQL ha demostrado ser un gestor de bases de datos de código abierto muy avanzado, ofreciendo control de concurrencia multiversión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario. La utilización de PostgreSQL como SGBD de un una Web obedece al propósito de la elaboración de un sistema con gran robustez y alto nivel de escalabilidad. PostgreSQL, posee muchas características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle.

Características:

- ✓ Altamente Extensible: Soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.

- ✓ Velocidad de respuesta**:** La velocidad de respuesta que ofrece este SGBD con bases de datos relativamente pequeñas pueden parecer un poco deficiente, pero esta misma velocidad la mantiene al gestionar bases de datos realmente grandes.
- ✓ Se caracteriza por ser un sistema estable, de alto rendimiento y gran flexibilidad.

1.9. Conclusiones parciales

En este capítulo se define el concepto de portal, las diferencias entre un sitio web y un portal web; se caracterizan las tendencias actuales y se realiza un estudio de los portales de calidad existentes tanto en el mundo como en la universidad.

Como resultado del análisis de los portales de calidad existentes, se pudo concluir que ninguno de estos portales responden a las características que requiere el portal del Grupo de Calidad de la Facultad 9, por lo que se necesita diseñar un nuevo portal que dinamice los procesos con las herramientas seleccionadas las cuales tuvieron la fundamentación del porqué de su selección.

Capítulo 2. Construcción de la aplicación

2.1 Introducción

A lo largo de este capítulo se realiza un análisis crítico y valorativo del análisis y diseño realizado y entregado por el analista. Se hace un análisis de la complejidad y funcionamiento de los algoritmos más importantes para la implementación del sistema, así como las estructuras de datos utilizadas para manejar los datos en el programa. Además se presenta la descripción principales métodos que le dan solución al problema planteado**.**

2.2 Valoración crítica del diseño propuesto por el analista

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura Modelo-Vista-Controlador (en lo adelante MVC) el cual divide cualquier aplicación que haga uso de él en tres partes o niveles:

- El Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista: transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados 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. Si por ejemplo, una misma aplicación debe ejecutarse tanto en un navegador estándar como en 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.

La implementación de una arquitectura MVC en un lenguaje de programación no orientado a objetos puede encontrarse con problemas de espacios de nombre y código duplicado, dificultando la lectura del

código de la aplicación. La orientación a objetos permite a los desarrolladores trabajar con objetos de la vista, objetos del controlador y clases del modelo, transformando las funciones en métodos. Se trata de un requisito obligatorio para las arquitecturas de tipo MVC.

Del diseño propuesto por el analista se pueden identificar los requisitos funcionales a implementar para que el sistema funcione correctamente, y realice las acciones que darán solución al problema planteado. Este diseño se examinó minuciosamente comprobando que las funcionalidades así como las clases persistentes relacionadas con estas se encuentren bien elaboradas para que esto facilite la implementación de las mismas. Mediante la descripción detallada de los casos de uso, se puede establecer una estrategia de trabajo para guiar la implementación de la aplicación, teniendo paso por paso lo que esta debe hacer.

Los requisitos funcionales a implementar son los siguientes:

RF1. Autenticar usuario

RF2. Crear planilla de evaluación de procedimiento de inicio

> **RF2.1.** Adicionar proyecto certificado (si el proyecto cumple con todo lo verificado y obtiene evaluación de certificado, el sistema dará la opción de adicionar el proyecto a la lista de los proyectos certificados).

RF3. Gestionar caso de prueba.

RF4. Foro

2.3 Descripción de los algoritmos no triviales a implementar

2.3.1 Algoritmo que permite al usuario autenticarse en el sistema (Anexo 1)

Pasos del algoritmo de la función autenticarse:

1. Se crea una consulta que será utilizada con posterioridad.
2. Se comprueba que la acción haya sido llamado por el método Post para saber que esta llamada procede del formulario.

3. En caso de que se cumpla lo anterior se procede a recoger los datos enviados.
4. Con los datos recogidos se realiza una búsqueda en la base de datos con el objetivo de comprobar si el usuario ya existe en la misma y tiene asignada la misma contraseña que se introdujo.
5. Si los datos del usuario y al contraseña coinciden entonces se cambia el atributo Authenticated, el atributo Attribute y la Credential del objeto de usuario antes creado.
6. Por último si los datos de la contraseña y del usuario recogidos del formulario no coinciden con ninguno de los que están guardados en la base de datos entonces se realiza una redirección para la página reg_user (página que brinda la opción de registrase en el sistema), para que con posterioridad se pueda realizar un registro exitoso.

2.3.2 Algoritmo que permite al usuario registrarse en el sistema (Anexo 2)

Pasos del algoritmo de la función Registrar usuario:

1. Se comprueba que los datos enviados por el formulario no estén vacios.
2. Si se cumple que ninguno de los datos están vacíos entonces se recogen todos los datos en variables para pasarles estas a la función Insertar de la clase UsuarioTable.
3. Se le pasan las variables a la función Insertar, pasándole el valor lector a la variable rol, esto se realiza de esta forma porque el usuario que se registre por primera vez a la aplicación va a tener este rol, con los accesos a paginas que posee dicho rol.

2.4 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 3 documentos, los cuales se referencian a continuación:

http://chileforge.cl/docman/view.php/81/108/estandar_de_codificacion.pdf
http://progra.iteso.mx/estandares/estandar%20codificacion%20c++/estandarcodificacion.pdf
http://www.chubut.gov.ar/informatica/docs/EstandaresCodificacion.pdf

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.

1.1 Indentación.

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

Líneas

Cuando las líneas 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
```

1.2 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.
    }*/
```

1.3 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
```

1.4 Espacios en blanco.

1.4.1 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).

1.4.2 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);

1.5 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.

1.6 Uso de las llaves

Se debe utilizar una línea para cada llave distinta de la primera.

Ejemplo:

```
public bool Buscarpersona (persona p) {
for (exp1; exp2; exp3) {
            if (exp) {
                        Código;
                    }
       }
     return var;
  }
```

2.5 Tratamiento de errores

El portal posee tratamiento de errores robusto proporcionado por mecanismos que posee el Framework usado, debido a que este permite crear los formularios de forma automática, con los campos que estos requieren y a la vez los comprueba para que los datos introducidos sean los correctos, mostrando mensajes de error para los campos en los que se ha introducido un valor no válido o en los campos donde

el dato es requerido y el usuario no lo ha introducido. Además de la evaluación de los errores en el servidor, se realiza validación del lado del cliente, a través de código Java Script.

2.6 Seguridad

2.6.1 Seguridad en la acción

La posibilidad de ejecutar una acción puede ser restringida a usuarios con ciertos privilegios. Las herramientas proporcionadas por Symfony para este propósito permiten la creación de aplicaciones seguras, en las que los usuarios necesitan estar autenticados antes de acceder a alguna característica o a partes de la aplicación. Añadir esta seguridad a una aplicación requiere dos pasos: declarar los requerimientos de seguridad para cada acción y autenticar a los usuarios con privilegios para que puedan acceder estas acciones seguras.

2.6.2 Restricción de acceso

Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida. En Symfony, los privilegios están compuestos por dos partes:

- ✓ Las acciones seguras requieren que los usuarios estén autenticados.
- ✓ Las credenciales son privilegios de seguridad agrupados bajo un nombre y que permiten organizar la seguridad en grupos.

Para restringir el acceso a una acción se crea y se edita un archivo de configuración YAML llamado security.yml en el directorio config/ del módulo. En este archivo, se pueden especificar los requerimientos de seguridad que los usuarios deberán satisfacer para cada acción o para todas las acciones.

2.7 Modelo de implementación

Luego de haber descrito las clases, atributos y sus relaciones se puede comenzar la implementación del sistema. La construcción del sistema debe ser en todo momento consistente con la documentación creada por el Analista y con la base de datos creada por el diseñador de Base de Datos, es ahora donde se construye el sistema en términos de componentes: ejecutables, ficheros de código fuente y scripts. El objetivo principal es desarrollar la arquitectura y definir la organización del código. Durante esta etapa se

obtiene el Modelo de Implementación que relaciona componentes y subsistemas. Para representar este tipo de modelo se emplea el Diagrama de Componentes.

2.7.1 Diagrama de componentes

El diagrama de componentes muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia. Se utiliza para modelar la vista estática del sistema a implementar. Muestra la organización y las dependencias lógicas entre los conjuntos de componentes de software.

Diagrama de componentes. (Anexo 3)

En la imagen que muestra el diagrama de componentes se pueden observar los paquetes que representan los archivos físicos donde se encuentran los elementos que le dan solución a la problemática planteada. En el paquete doctrine se encuentran los elementos relacionados con el modelo de la aplicación; en los paquetes modules y controladores frontales se encuentran los elementos relacionados con los controladores de la aplicación. En el paquete template se encuentran los elementos relacionados con las vistas de la aplicación la que es mostrada al usuario que interactúa con el portal, en este paquete se encuentra el archivo de la vista en el cual se cargan los demás archivos relacionados con las vistas que se encuentran dentro de los paquetes inicioproyecto y casoprueba. En el paquete Componentes de Symfony se encuentran los elementos y archivos relacionados con el Framework de desarrollo utilizado para formular la solución.

2.8 Funciones u operaciones necesarias

2.8.1 Función que permite insertar una planilla de inicio de proyecto (Anexo 4)

Pasos del algoritmo de la función crear planilla de inicio de proyecto:

1. Se comprueba que la acción haya sido llamada por el método Post para saber que esta procede del formulario.
2. Se valida que los campos requeridos no hayan sido enviados vacios.
3. En caso de que se cumpla lo anterior se procede a recoger los datos enviados.

4. Después de realizada la acción de recoger los datos se procede a llamar al método Insertar de la clase InicioTable para que inserte los datos correspondientes.

2.8.2 Función que permite crear un caso de prueba de un proyecto (Anexo 5)

Pasos del algoritmo de la función crear planilla de inicio de proyecto:

1. Se comprueba que la acción haya sido llamada por el método Post para saber que esta procede del formulario.
2. Se verifica que no exista ninguna versión de una prueba a un proyecto repetida.
3. Después de realizada esta verificación se pasa la recogida de los datos en las variables correspondientes.
4. Con las variables recogidas se procede a llamar al método encargado de insertar el caso de prueba pasándole los parámetros correspondientes.

2.8.3 Función que permite modificar una planilla de inicio de proyecto

Pasos del algoritmo de la función modificar una planilla de inicio de proyecto:

1. Se comprueba que la acción haya sido llamada por el método Post para saber que esta procede del formulario correspondiente a ella.
2. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla Inicio donde coincida que el id del proyecto sea igual al dato que viene del formulario.
3. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla ProyectosExisten donde coincida que el nombre del proyecto sea igual al dato que viene del formulario.
4. Si el valor del campo certificado es true entonces se procede a cambiar el valor del campo activo en la tabla ProyectosExisten por true y el valor del campo certificado en la tabla Inicio por true.
5. Se cambian todos los valores de los campos de la tabla Inicio por los que se introdujeron en el formulario por el usuario.

2.8.4 Función que permite eliminar una planilla de inicio de proyecto

Pasos del algoritmo de la función crear planilla de inicio de proyecto:

1. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla Inicio donde coincida que el id del proyecto y la versión del mismo sean iguales a los datos que vienen del formulario.
2. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla ProyectosExisten donde coincida que el nombre del proyecto sea igual al dato que viene del formulario.
3. Se precede a cambiar el campo activo en la tabla ProyectosExisten por el valor false.
4. Se precede a eliminar la planilla de inicio de proyecto.

2.8.5 Función que permite modificar los datos iniciales de un caso de prueba de un proyecto

Pasos del algoritmo de la función crear planilla de inicio de proyecto:

1. Se comprueba que la acción haya sido llamada por el método Post para saber que esta procede del formulario correspondiente a ella
2. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla Pruebas donde coincida que el nombre del proyecto y la versión de la prueba del mismo sea igual a los datos que vienen del formulario.
3. Se procede a actualizar cada uno de los campos de la tupla seleccionada con los valores entrados por el formulario.

2.8.6 Función que permite eliminar un caso de prueba de un proyecto

Pasos del algoritmo de la función crear planilla de inicio de proyecto:

1. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla Pruebas donde coincida que el nombre del proyecto y la versión de la prueba del mismo sea igual a los datos que vienen del formulario.

2. Se realiza una consulta a la BD donde se selecciona la tupla de la tabla Pruebas donde coincida que el nombre del proyecto y la versión de la prueba del mismo sea igual a los datos que vienen del formulario.

3. Se procede a eliminar la versión del caso de prueba del proyecto en cascada.

2.9 Conclusiones parciales

En este capítulo se realizó un análisis del diseño entregado por el analista donde se hicieron algunas modificaciones y aportes necesarios en el proceso de implementación para obtener los resultados deseados. Se hizo un esbozo de los componentes utilizados en la solución de las funcionalidades principales. Se logró establecer una línea base sencilla que permita la programación eficiente por parte del desarrollador. Se explicó la implementación de aquellas partes que pudieran resultar confusas, ambiguas o complicadas.

Capítulo 3. Validación de la solución propuesta

3.7.1 Introducción

En este capítulo se aborda el tema referente a la realización de las pruebas; se realiza un análisis del tipo de prueba a utilizar para validar el software y el diseño de casos de prueba. En este capítulo también se crean las planillas de los casos de prueba para cada los casos de uso que se han implementado donde se detallan los mismos con sus descripciones generales, condiciones de ejecución, y secciones a probar en los casos de uso.

3.2 Pruebas de software

La calidad de un sistema está determinada, entre otras cosas, por la coincidencia entre lo que se programó y los requisitos establecidos en la primera fase. Para comprobar el grado de cumplimiento de estos requisitos se usan las pruebas del sistema. Estas definen un conjunto amplio de acciones de comprobación que abarcan todas las características que determinan la calidad de un software. Se comprueban las funcionalidades diseñando casos de prueba que definen cómo proceder. Estos casos de prueba incluyen los juegos de datos a usar que son los válidos o esperados y los no válidos o no esperados por el programa. Además establecen los resultados a alcanzar en correspondencia de la lógica del programa y los datos ingresados. Describen las condiciones generales en las que se debe aplicar las pruebas para obtener los objetivos propuestos. El objetivo de los casos de prueba es forzar al máximo el sistema en los puntos críticos para encontrar fallos y detectar defectos. Las pruebas se deben aplicar durante todo el ciclo de vida del software e invariablemente se le debe dedicar una gran parte del esfuerzo total del desarrollo. Se deben planificar correctamente desde el inicio y establecer qué hacer, cómo hacer, quién va a hacer y en qué condiciones hacer las comprobaciones. Es beneficioso que los desarrolladores prueben su producto pero que no falte la mano de terceras personas que no intervinieron en el proyecto directamente ya que así se detecta mayor cantidad de fallas. (S: Pressman)

A la hora de realizarle una prueba a un producto de software se debe escoger los tipos de prueba que se adapte mejor al sistema que se va a probar. Por lo que se debe tener en cuenta el lenguaje de programación utilizado, así como el proceso de desarrollo, las características de los desarrolladores, la

plataforma en que se ejecutan los procesos, los errores más importantes, el tipo de aplicación implementada y si realiza conexiones a bases de datos o no.

3.3 Pruebas de caja blanca

Las pruebas de Caja Blanca se nombran de esta forma porque a diferencia de las pruebas de Caja Negra que actúan sobre la interfaz, estas revisan la parte interna del software, específicamente sobre el código fuente. Se basan en el examen minucioso de los detalles procedimentales. Se comprueban los caminos lógicos del sistema generando casos de prueba que ejerciten las estructuras condicionales y los bucles. Existen varios métodos que analizan diferentes partes del programa y se complementan entre sí para garantizar la calidad del sistema.

La técnica del camino básico se utiliza para comprobar la complejidad lógica de un diseño procedimental, permite diseñar casos de prueba para cubrir todas las sentencias de un programa a partir de la obtención de un conjunto de caminos independientes. La complejidad ciclomática, como resultado fundamental de estas pruebas, acota la cantidad mínima de casos de prueba que se deben ejecutar. La prueba de las Condiciones es un método que se encamina hacia la ejercitación de las condiciones. Se basa en el principio de que si un conjunto de casos de prueba es capaz de ejercitar todas las condiciones contenidas en un bloque de código, este mismo conjunto serviría para encontrar más errores en el programa que no tengan que ver directamente con las condiciones.

La prueba del Flujo de Datos verifica la validez en el uso de las variables para manipular los datos de la aplicación. Selecciona los casos de prueba atendiendo a las definiciones y los usos de las variables. El procedimiento indica que se debe encontrar las sentencias donde se define cada variable y las sentencias donde se hace uso de las mismas. Luego se encuentran las cadenas de definición, uso que representan el ciclo de vida de las variables y se diseñan casos de prueba que las ejecuten en su totalidad. La prueba de los bucles se centra en la validez de las estructuras cíclicas o bucles. El objetivo es probar el comportamiento de estas estructuras en sus valores límites de iteración. Los bucles se clasifican en cuatro tipos: Bucles simples, Bucles anidados, Bucles concatenados y Bucles no estructurados. Aunque la esencia es la misma, cada tipo se prueba de forma diferente. De lo anterior pudiera pensarse que las pruebas de Caja Blanca logran enmendar todos los errores del programa. La desventaja de estas es entonces de tipo logística ya que resulta imposible abarcar todo el código fuente de un sistema

medianamente grande. El tiempo necesario para realizarlas sería considerable y se torna compleja su aplicación sobre algoritmos críticos.

3.4 Pruebas de caja negra

Las Pruebas de Caja Negra deben su nombre a los elementos que estas revisan y las condiciones en que se hace la revisión. Estas se basan en los requerimientos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Este tipo de prueba es importante a la hora de medir el grado de cumplimiento de los requerimientos solicitados por el cliente y se aplican sobre la interfaz de la aplicación observando las respuestas del sistema antes determinadas acciones y los datos de salida para determinados datos de entrada. (S: Pressman)

Al analizar los métodos de prueba de Caja Blanca y Caja Negra se llega a la conclusión de que es más factible aplicar las pruebas de Caja Negra para comprobar la validez en las respuestas del programa ante cada una de las acciones que realiza el usuario y las respuesta que da como salida el sistema a cada una de ellas. Para probar la validez de la solución propuesta en la implementación del sistema se decidió probar los elementos fundamentales de la interfaz de usuario mediante casos de prueba de Caja Negra así como la validez de las salidas del sistema en dependencia de las entradas del usuario.

3.5 Pruebas unitarias y funcionales

Las pruebas unitarias permiten probar, como su nombre lo indica, cada unidad independiente del software. Actúan esencialmente sobre el código fuente y sobre los elementos básicos de la interfaz de cada módulo. Pressman (S. Pressman) plantea que los casos de prueba que se generan durante las pruebas de unidad deben estar encaminados a verificar los siguientes elementos:

- ✓ **Interfaz**: “Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada”.
- ✓ **Estructuras de datos locales**: “Se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente, conservan su integridad durante todos los pasos de ejecución del algoritmo”.
- ✓ **Condiciones límites**: “Se prueban las condiciones límites para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento”.

- ✓ **Caminos independientes**: “Se ejercitan todos los caminos independientes (caminos básicos) de la estructura de control con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez”.
- ✓ **Caminos de manejo de errores**: “Se prueban todos los caminos de manejo de errores”.

Con las pruebas unitarias es posible aislar una parte del código de manera que pueda ser analizado. Un ejemplo de esto es evaluar las funciones o métodos, a los cuales se les realiza una entrada de datos para obtener los datos de salida correctos. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular. Las pruebas funcionales no solo validan la transformación de una entrada en una salida, sino que validan una característica completa. Por ejemplo un sistema de cache solamente puede ser validado por una prueba funcional, ya que comprende más de 1 solo paso: la primera vez que se solicita una página, se produce su código; la segunda vez se obtiene directamente de la cache. De modo que las pruebas funcionales validan procesos y requieren de un escenario. En Symfony se deberían crear pruebas funcionales para todas las acciones.

Estas pruebas simulan la navegación del usuario, realizan peticiones y comprueban los elementos de la respuesta tal y como lo haría manualmente un usuario para validar que una determinada acción hace lo que se supone que debe hacer. En las pruebas funcionales se ejecuta un escenario correspondiente a lo que se denomina un caso de uso.

3.6 Diseño de los casos de prueba

Debido a que el diseño de casos de prueba para un producto de software requiere tanto esfuerzo como el propio diseño del mismo, los diseñadores o ingenieros del software a menudo no le prestan la importancia que esto requiere y desarrollan casos de prueba que parecen ser adecuados pero que tienen poca garantía de ser completos para evitar esto a la hora de realizar las pruebas se debe recordar el objetivo principal de las mismas que es diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con las mínima cantidad de esfuerzo y tiempo posible. Como bien dice Tsuneo Yamaura: “Existe una sola regla para el diseño de casos de prueba: cubrir todas las posibilidades, sin hacer demasiados casos de prueba.”

Con el objetivo de probar el correcto funcionamiento de la aplicación se crean los diseños de casos de prueba correspondientes. A continuación en forma de anexos se relacionan algunos de los diseños casos de prueba implementados:

Diseño de Caso de Pruebas funcionales basados en CU para el caso de uso Crear planilla de evaluación de procedimiento de inicio.

1. Descripción General.

El caso consiste en que el iniciador debe revisar la documentación que provee el jefe del proyecto y verificar el cumplimiento de varias tareas, posteriormente el iniciador emite la evaluación del proyecto que puede ser de (certificado o no certificado). Si el proyecto se evalúa de certificado se podrá adicionar a la lista de proyectos certificados.

2. Condiciones de Ejecución.

El sistema debe estar instalado y ejecutándose correctamente. El actor debe estar autenticado con los permisos necesarios.

3. Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Crear planilla de evaluación de procedimiento de inicio.	EC 1.1: Crear planilla de evaluación de procedimiento de inicio.	El sistema muestra un formulario con todos los campos a llenar.
	EC 1.2: Introducir el nombre del autor.	
	EC 1.3: Dejar el campo nombre del autor vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.4: Dato nombre del autor incorrecto	El sistema muestra un mensaje de error al lado del campo.
	EC 1.5: Introducir el nombre del proyecto.	

EC 1.6: Dejar el campo nombre del proyecto vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.7: Dato nombre del proyecto incorrecto	El sistema muestra un mensaje de error al lado del campo.
EC 1.8: Introducir el número de la versión.	
EC 1.9: Dejar el campo número de la versión vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.10: Dato número de la versión incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.11: Introducir cual metodología se usa.	
EC 1.12: Dejar el campo de cual metodología se usa vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.13: Dato cual metodología se usa incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.14: Introducir la iteración.	
EC 1.15: Dejar el campo iteración vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.16: Dato iteración incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.17: Introducir las herramientas para gestión de riesgos.	

	EC 1.18: Dejar el campo herramientas para gestión de riesgos vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.19: Dato herramientas para gestión de riesgos incorrecto.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.17: Introducir precio del producto.	
	EC 1.18: Dejar el campo precio del producto vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.19: Dato precio del producto incorrecto.	El sistema muestra un mensaje de error al lado del campo.

4. Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1.	Nombre del autor	Lista desplegable	No	En este campo se puede introducir cualquier tipo de dato.
2.	Nombre del proyecto	Campo de texto	No	En este campo se puede introducir cualquier tipo de dato.
3.	Versión	Campo de texto	No	Este campo debe contener números enteros.
4.	Fecha de asignados los recursos	Campo de selección	No	Se debe seleccionar una fecha.
5.	Metodología	Campo de selección	No	

6.	¿Cuál?	Campo de texto	No	En este campo se puede introducir cualquier tipo de dato.
7.	Plan de capacitación	Campo de selección	No	
8.	Cronograma del proyecto	Campo de selección	No	
9.	Iteración	Campo de texto	No	Este campo debe contener números enteros.
10.	Lista de riesgos	Campo de selección	No	
11.	Plan de mitigación de riesgos	Campo de selección	No	
12.	Plan de contingencia	Campo de selección	No	
13.	Documento visión	Campo de selección	No	
14.	Herramientas para gestión de riesgos	Campo de texto	No	
15.	Activ. de eval. y gestión de riesgos en cronograma (alto)	Campo de selección	No	
16.	Cumplido el plazo	Campo de selección	No	
17.	Definido un rol económico	Campo de selección	No	
18.	Precio del producto	Campo de texto	No	Este campo debe contener números reales.
19.	Registrados costos de mat. primas y materiales	Campo de selección	No	

20.	Definida la técnica de estimación a utilizar	Campo de selección	No	
21.	Certificado	Campo de selección	No	

El caso de prueba se encuentra en su totalidad en el siguiente anexo **(Anexo 7)**

Diseño de Caso de Pruebas funcionales basados en CU para el caso de uso Gestionar caso de prueba.

1. Descripción General.

Se inicia cuando el probador entra al modulo de prueba y selecciona la opción de gestionar caso de prueba.

2. Condiciones de Ejecución.

El sistema debe estar instalado y ejecutándose correctamente. El actor debe estar autenticado con los permisos necesarios.

3. Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Crear caso de prueba	EC 1.1: Crear caso de prueba.	El sistema muestra un formulario con todos los
	EC 1.2: Introducir el nombre del proyecto.	
	EC 1.3: Dejar el campo nombre del proyecto vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.4: Dato nombre del proyecto incorrecto.	El sistema muestra un mensaje de error al lado del campo.

EC 1.5: Introducir la versión.	
EC 1.6: Dejar el campo versión vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.7: Dato versión incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.8: Introducir el solapin del autor.	
EC 1.9: Dejar el campo solapin del autor vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.10: Dato solapin del autor incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.11: Introducir la descripción.	
EC 1.12: Dejar el campo descripción vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.13: Introducir la descripción general.	
EC 1.14: Dejar el campo descripción general vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.15: Introducir las condiciones de ejecución.	
EC 1.16: Dejar el campo condiciones de ejecución vacío.	El sistema muestra un mensaje de error al lado

	del campo.
EC 1.17: Introducir el nombre de la sección.	
EC 1.18: Dejar el campo nombre de la sección vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.19: Introducir el nombre de la sección.	
EC 1.20: Dejar el campo nombre de la sección vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.21: Introducir el escenario de la sección.	
EC 1.22: Dejar el campo escenario de la sección vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.23: Introducir la descripción de la funcionalidad.	
EC 1.24: Dejar el campo descripción de la funcionalidad vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.25: Introducir el flujo central.	
EC 1.26: Dejar el campo el flujo central vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.27: Introducir el escenario.	

EC 1.28: Dejar el campo escenario vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.29: Introducir las variables.	
EC 1.30: Dejar el campo variables vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.31: Introducir las respuestas del sistema.	
EC 1.32: Dejar el campo respuestas del sistema vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.33: Introducir los resultados de la prueba.	
EC 1.34: Dejar el campo resultados de la prueba vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.35: Introducir el número.	
EC 1.36: Dejar el campo número vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.37: Dato número incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.38: Introducir el nombre del campo.	
EC 1.39: Dejar el campo nombre del	El sistema muestra un

campo vacío.	mensaje de error al lado del campo.
EC 1.40: Dato nombre del campo incorrecto.	El sistema muestra un mensaje de error al lado del campo.
EC 1.38: Introducir la clasificación.	
EC 1.39: Dejar el campo clasificación vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.40: Dato nombre del campo clasificación.	El sistema muestra un mensaje de error al lado del campo.
EC 1.41: Introducir la clasificación.	
EC 1.42: Dejar el campo clasificación vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.43: Dato nombre del campo clasificación.	El sistema muestra un mensaje de error al lado del campo.
EC 1.41: Introducir la descripción.	
EC 1.42: Dejar el campo descripción vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 1.43: Introducir la no conformidad.	
EC 1.44: Dejar el campo no conformidad	El sistema muestra un

	vacío.	mensaje de error al lado del campo.
	EC 1.45: Introducir el aspecto correspondiente.	
	EC 1.46: Dejar el campo aspecto correspondiente vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.47: Dato aspecto correspondiente incorrecto.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.48: Introducir la etapa de detección.	
	EC 1.49: Dejar el campo etapa de detección vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.50: Dato etapa de detección incorrecto.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.51: Introducir la recomendación.	
	EC 1.52: Dejar el campo recomendación vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 1.53: Introducir la respuesta del equipo.	
	EC 1.54: Dejar el campo respuesta del equipo vacío.	El sistema muestra un mensaje de error al lado del campo.

SC 2: Eliminar caso de prueba	EC 2.1: Eliminar caso de prueba.	El sistema muestra un formulario con todos los campos a seleccionar.
SC 3: Modificar caso de prueba	EC 3.1: Modificar caso de prueba.	El sistema muestra un formulario con todos los campos a llenar.
	EC 3.2: Introducir el nombre del proyecto.	
	EC 3.3: Dejar el campo nombre del proyecto vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.4: Dato nombre del proyecto incorrecto.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.5: Introducir la versión.	
	EC 3.6: Dejar el campo versión vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.7: Dato versión incorrecto.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.8: Introducir el solapin del autor.	
	EC 3.9: Dejar el campo solapin del autor vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.10: Dato solapin del autor	El sistema muestra un

incorrecto.	mensaje de error al lado del campo.
EC 3.11: Introducir la descripción.	
EC 3.12: Dejar el campo descripción vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 3.13: Introducir la descripción general.	
EC 3.14: Dejar el campo descripción general vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 3.15: Introducir las condiciones de ejecución.	
EC 3.16: Dejar el campo condiciones de ejecución vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 3.17: Introducir el nombre de la sección.	
EC 3.18: Dejar el campo nombre de la sección vacío.	El sistema muestra un mensaje de error al lado del campo.
EC 3.19: Introducir el nombre de la sección.	
EC 3.20: Dejar el campo nombre de la sección vacío.	El sistema muestra un mensaje de error al lado del campo.

	EC 3.21: Introducir el escenario de la sección.	
	EC 3.22: Dejar el campo escenario de la sección vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.23: Introducir la descripción de la funcionalidad.	
	EC 3.24: Dejar el campo descripción de la funcionalidad vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.25: Introducir el flujo central.	
	EC 3.26: Dejar el campo el flujo central vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.27: Introducir el escenario.	
	EC 3.28: Dejar el campo escenario vacío.	El sistema muestra un mensaje de error al lado del campo.
	EC 3.29: Introducir las variables.	
	EC 3.30: Dejar el campo variables vacío.	El sistema muestra un mensaje de error al lado del campo.
SC 4: Mostrar caso de prueba	EC 1.1: Mostrar caso de prueba.	El sistema muestra un formulario con los campos a seleccionar.

4. Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1.	Nombre del proyecto.	Lista desplegable	No.	En este campo se puede introducir cualquier tipo de dato.
2.	Versión.	Campo de texto.	No.	Este campo debe contener números enteros.
3.	Solapin del autor.	Lista desplegable	No.	Este campo debe contener números enteros.
4.	Descripción.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de dato
5.	Descripción general.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
6.	Condiciones de ejecución.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
7.	Nombre de la sección.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
8.	Escenario de la sección.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
9.	Descripción de la funcionalidad	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
10.	Flujo central	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de

11.	Escenario	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
12.	Variables	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
13.	Respuesta del sistema	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
14.	Resultado de la prueba	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
15.	Número	Campo de texto.	No.	Este campo debe contener números enteros.
16.	Nombre del campo.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
17.	Clasificación.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
18.	Descripción.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
19.	No conformidad.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
20.	Aspecto correspondiente	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
21.	Etapa de detección.	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
22.	Recomendación	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de
23.	Respuesta del equipo	Campo de texto.	No.	En este campo se puede introducir cualquier tipo de

El caso de prueba se encuentra en su totalidad en el siguiente anexo **(Anexo 8)**

En estos casos de prueba se probaron todas y cada una de las variantes en que pueden ser entrados los datos al sistema para ver si la respuesta de este es la que deber ser ante cada una de estas situaciones. Las variantes de entrada de datos pueden ser: datos incorrectos, que es cuando en un campo se debe escribir un tipo de datos ya sea un número, una fecha o correo y se le inserta otro; datos vacíos que es cuando se deja el campo vacío sin entrar ningún dato.

Los diseños de casos de prueba arrojaron varios resultados de los cuales el más destacado es que la aplicación no posee no conformidades, por lo que se asegurar que la misma cumple con los requisitos planteados por el analista, además de que responde correctamente a cada una de las condiciones por las que puede pasar el usuario en su navegación por la misma.

3.7 Conclusiones parciales

En este capítulo se realizó un análisis de lo que es una prueba de software para con este conocimiento pasar a escoger la prueba que más se ajusta al sistema desarrollado. Se analizan las pruebas de caja blanca y las pruebas de caja negra siendo la última la seleccionada para ponerse en práctica en la validación de la aplicación para de esta forma determinar si la misma cumple con los parámetros que plantea el analista del sistema así como su cumplimiento con los requisitos funcionales del software. Por último se crean los diseños casos de prueba basados en casos de uso de los módulos implementados.

Conclusión

Una vez finalizada la investigación se ha tenido a bien resaltar una serie de conclusiones las cuales se enumeran a continuación.

- ✓ El diseño propuesto por el analista está claro, simple y orientado al implementador.
- ✓ La utilización del Framework Symfony permitió disminuir el tiempo empleado en la codificación, y estandarizar la propuesta con el resto de los módulos que componen el Portal.
- ✓ Los estilos y estándares de implementación potencian la organización, la legibilidad y reusabilidad del código implementado.
- ✓ Todos los requisitos funcionales previstos para esta versión del portal fueron implementados.
- ✓ Se logró la implementación del portal de calidad de la facultad 9 que permite agilizar el trabajo relacionado con los procedimientos de Inicio de Proyecto y Estrategia de Prueba.
- ✓ Los Casos de Prueba diseñados permiten detectar errores tempranos, agilizar y simplificar el flujo de trabajo de pruebas.
- ✓ Se obtuvo la documentación técnica apropiada, generada por el implementador en el desarrollo de su rol.

Recomendaciones

La presente investigación tiene como recomendaciones para las próximas versiones de la misma las que se enumeran a continuación:

- ✓ Incorporar la autenticación al portal mediante el dominio UCI.
- ✓ Incorporar un buscador al portal creado.
- ✓ Realizar casos de prueba de caja blanca para la validación de la aplicación creada.

Anexos.

Anexo 1

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

  if ($this->getRequest()->getMethod() == sfRequest::POST)
          {
        $user = $this->getRequestParameter('user');
        $pass = $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());
               $entro = true;
             }
          }
         }
        endforeach;
      }
      if(!$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');
         }
 }
```

Anexo 2

```
public function executeReg_user(sfWebRequest $request)
  {

     if  ($this->getRequest()->getMethod() == sfRequest::POST)
           {
        //$ci = $this->getRequestParameter('ci');
        $nombre = $this->getRequestParameter('nombre');
        $apellidos = $this->getRequestParameter('apellidos');

        $solapin = $this->getRequestParameter('solapin');

        $sol = Doctrine::getTable('Usuario')
          ->createQuery('b')
          ->where('b.solapin = ?', $solapin)
          ->execute();

        if($sol[0]->getSolapin() != null)
          {
             $this->getUser()->setFlash('error', sprintf('El solapin '.$this-> getRequestParameter('solapin').' ya existe'));
             $this->redirect('inicio/reg_user');
          }

        $user = $this->getRequestParameter('usuario');

        $usuario = Doctrine::getTable('Usuario')
          ->createQuery('b')
          ->where('b.usuario= ?', $user)
          ->execute();

        if($usuario[0]->getUsuario() != null)
          {
             $this->getUser()->setFlash('error', sprintf('El usuario '.$this-> getRequestParameter('usuario').' ya existe'));
             $this->redirect('inicio/reg_user');
          }

        $contrasenna = $this->getRequestParameter('contrasenna');

        $correo = $this->getRequestParameter('correo');

        $corr = Doctrine::getTable('Usuario')
          ->createQuery('b')
          ->where('b.correo= ?', $correo)
          ->execute();

        if($corr[0]->getUsuario() != null)
          {
```

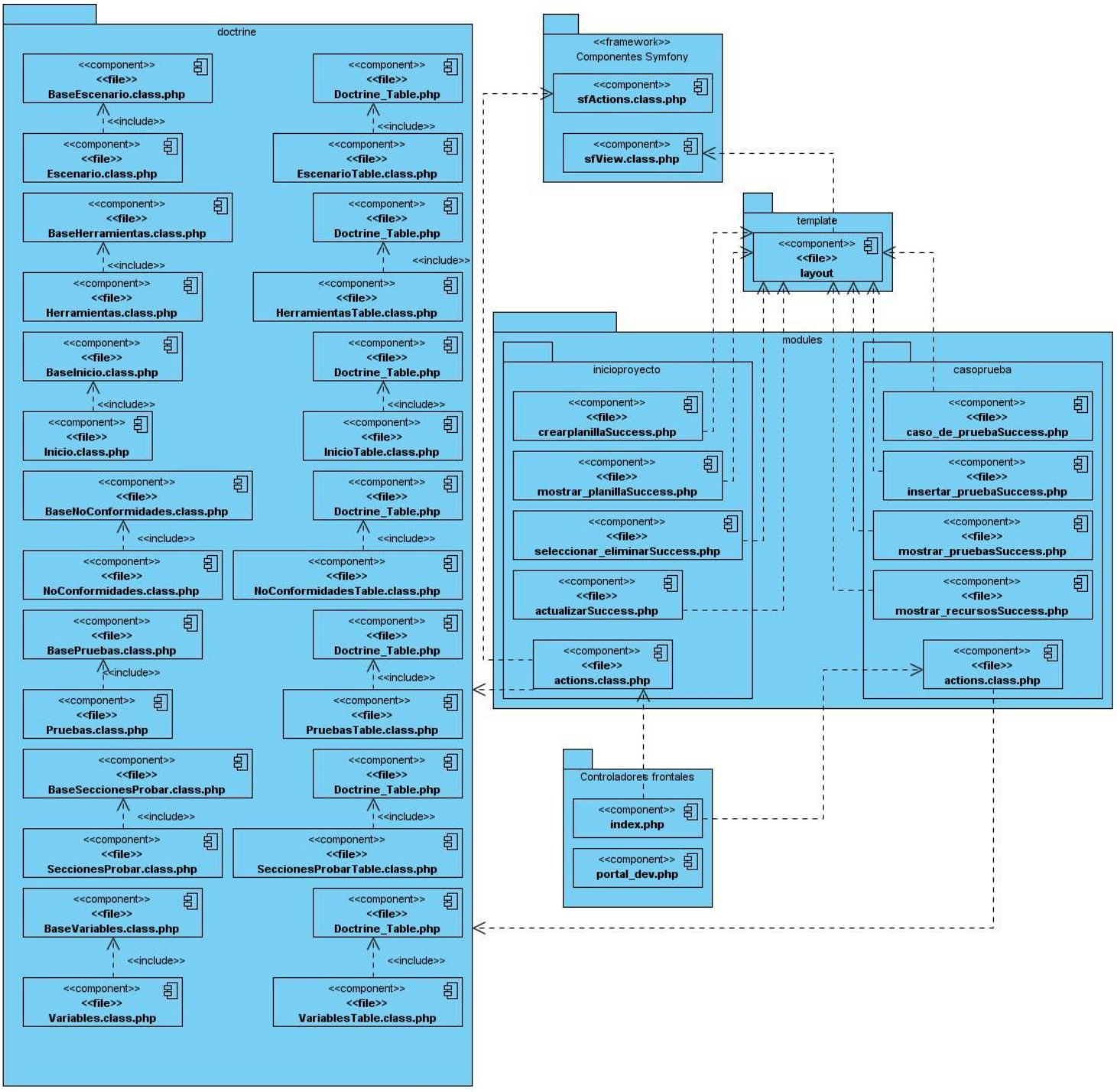
```
            $this->getUser()->setFlash('error', sprintf('El correo '.$this-> getRequestParameter('correo').' ya esta en uso'));
            $this->redirect('inicio/reg_user');
          }

        $rol = "lector";
        $fecha = date('d/m/y');

        UsuarioTable::Insertar($nombre,$apellidos,$solapin,$correo,$user,$contrasenna,$fecha);
        UserRolTable::Insertar($rol, $solapin);
        $this->getUser()->setFlash('notice', sprintf('Se he registrado correctamente'));
        $this->redirect('inicio/index');

       }
  }
```

Anexo 3


doctrine
<<component>> <<file>> BaseEscenario.class.php
<<component>> <<file>> Doctrine_Table.php
<<include>>
<<component>> <<file>> Escenario.class.php
<<component>> <<file>> EscenarioTable.class.php
<<component>> <<file>> BaseHerramientas.class.php
<<component>> <<file>> Herramientas.class.php
<<component>> <<file>> HerramientasTable.class.php
<<component>> <<file>> BaseInicio.class.php
<<component>> <<file>> Inicio.class.php
<<component>> <<file>> InicioTable.class.php
<<component>> <<file>> BaseNoConformidades.class.php
<<component>> <<file>> NoConformidades.class.php
<<component>> <<file>> NoConformidadesTable.class.php
<<component>> <<file>> BasePruebas.class.php
<<component>> <<file>> Pruebas.class.php
<<component>> <<file>> PruebasTable.class.php
<<component>> <<file>> BaseSeccionesProbar.class.php
<<component>> <<file>> SeccionesProbar.class.php
<<component>> <<file>> SeccionesProbarTable.class.php
<<component>> <<file>> BaseVariables.class.php
<<component>> <<file>> Variables.class.php
<<component>> <<file>> VariablesTable.class.php
<<framework>> Componentes Symfony
<<component>> sfActions.class.php
<<component>> sfView.class.php
template
<<component>> <<file>> layout
modules
inicioproyecto
<<component>> <<file>> crearplanillaSuccess.php
<<component>> <<file>> mostrar_planillaSuccess.php
<<component>> <<file>> seleccionar_eliminarSuccess.php
<<component>> <<file>> actualizarSuccess.php
<<component>> <<file>> actions.class.php
casoprueba
<<component>> <<file>> caso_de_pruebaSuccess.php
<<component>> <<file>> insertar_pruebaSuccess.php
<<component>> <<file>> mostrar_pruebasSuccess.php
<<component>> <<file>> mostrar_recursosSuccess.php
<<component>> <<file>> actions.class.php
Controladores frontales
<<component>> index.php
<<component>> portal_dev.php

Anexo 4

```
public function executeCrearplanilla()
  {
     if ($this->getRequest()->getMethod() == sfRequest::POST)
           {
        $autor = $this->getRequestParameter('autor');
        $proyecto = Doctrine::getTable('Inicio')
                ->createQuery('a')
                ->where('a.nomb_proy = ?', $this->getRequestParameter('nomb_proy'))
                ->execute();

        if($proyecto[0]->getNombProy() == null)           {
           $nomb_proy = $this->getRequestParameter('nomb_proy');
         }
         else
           {
             $this->getUser()->setFlash('error', sprintf('La planilla para el proyecto '.$this-
>getRequestParameter('nomb_proy').' ya existe'));
             $this->redirect('inicioproyecto/crearplanilla');
           }
        $num_version = $this->getRequestParameter('num_version');


        if ($this->getRequestParameter('met')== 'on')
           {
             $met = true;
             $meto_entrar = $this->getRequestParameter('meto_entrar');
           }
         else
           {
             $met = false;
           }
        if ($this->getRequestParameter('plan')== 'on')
         {
           $plan = true;
         }
        else
         {
           $plan = false;
         }
        if ($this->getRequestParameter('cronograma')== 'on')
         {
          $cronograma = true;
           $num_iteracion = $this->getRequestParameter('num_iteracion');
         }
        else
         {
           $cronograma = false;
```

```
  }

if ($this->getRequestParameter('existe_lista')== 'on')
  {
    $existe_lista = true;
  }
else
  {
    $existe_lista = false;
  }
if ($this->getRequestParameter('existe_planmiti')== 'on')
  {
    $existe_planmiti = true;
  }
else
  {
    $existe_planmiti = false;
  }

if ($this->getRequestParameter('existe_plancon')== 'on')
  {
    $existe_plancon = true;
  }
else
  {
    $existe_plancon = false;
  }

if ($this->getRequestParameter('existe_docvision')=='on')
  {
    $existe_docvision = true;
  }
else
  {
    $existe_docvision = false;
  }
$nomb_herra = $this->getRequestParameter('nomb_herra');
$herram = Doctrine::getTable('Herramientas')
          ->createQuery('a')
          ->where('a.nombre = ?', $nomb_herra )
          ->execute();

if($herram[0]->getNombre() == null)
  {
     HerramientasTable::Insertar($nomb_herra);
  }

if ($this->getRequestParameter('eval_riesgos')== 'on')
  {
    $eval_riesgos = true;
```

```
  }
else
  {
    $eval_riesgos = false;
  }
if ($this->getRequestParameter('plazo_cumplido')== 'on')
  {
    $plazo_cumplido = true;
  }
else
  {
    $plazo_cumplido = false;
  }
if ($this->getRequestParameter('definido_rol')== 'on')
  {
    $definido_rol = true;
  }
else
  {
    $definido_rol = false;
  }
   $precio = $this->getRequestParameter('precio');

 if ($this->getRequestParameter('costo_proyecto')== 'on')
  {
    $costo_proyecto = true;
  }
else
  {
    $costo_proyecto = false;
  }

  if ($this->getRequestParameter('tecnica')== 'on')
  {
    $tecnica = true;
  }
else
  {
    $tecnica = false;
  }
 if ($this->getRequestParameter('certificado')== 'on')
  {
    $certificado = true;
  }
else
  {
    $certificado = false;
  }

$fecha = $this->getRequestParameter('fecha');
```

```
InicioTable::Insertar($autor,$nomb_proy,$num_version,$fecha,$met,$meto_entrar,$plan,$cronograma,$num_iteracion,$existe_lista,$existe_planmiti,$existe_plancon,$existe_docvision,$nomb_herra,$eval_riesgos,$plazo_cumplido,$definido_rol,$precio,$costo_proyecto,$tecnica,$certificado);

          $this->getUser()->setFlash('notice', sprintf('La planilla ha sido insertada correctamente'));
          $this->redirect('inicio/index');
        }
  }
```

Anexo 5

```
if  ($this->getRequest()->getMethod() == sfRequest::POST && $this->getRequestParameter('num1') == 1)
    {
       $nomb_proy = $this->getRequestParameter('nomb_proy');
       $num_version = $this->getRequestParameter('num_version');

         $val_prueba = Doctrine::getTable('Pruebas')
           ->createQuery('b')
           ->where('b.nombre_proy = ?', $this->getRequestParameter('nomb_proy'))
           ->andWhere('b.num_version = ?', $this->getRequestParameter('num_version'))
           ->execute();

                     if($val_prueba[0]->getAutor() != null)
                     {
$this->getUser()->setFlash('error', sprintf('La versión de prueba para este proyecto ya existe'));
                              $this->redirect('casoprueba/caso_de_prueba');
                     }

       $solapin = $this->getRequestParameter('solapin');
       $desc = $this->getRequestParameter('desc');
       $fecha = $this->getRequestParameter('fecha');
       $desc_general = $this->getRequestParameter('desc_general');
       $cond_ejec = $this->getRequestParameter('cond_ejec');

PruebasTable::Insertar($nomb_proy, $num_version, $cond_ejec,$desc, $fecha, $desc_general , $solapin);
       $this->getUser()->setFlash('notice', sprintf('La prueba ha sido insertada correctamente'));

           $pruebas = Doctrine::getTable('Pruebas')
          ->createQuery('b')
          ->where('b.nombre_proy = ?', $this->getRequestParameter('nomb_proy'))
          ->andWhere('b.num_version = ?', $this->getRequestParameter('num_version'))
          ->execute();

           $this->id_prueba = $pruebas[0]->getIdprueba();

    }
```

Glosario de términos

IDE: Entorno de Desarrollo Integrado.

IIS: Servidor de Información de Internet (Internet Information Server).

.NET: plataforma de desarrollo de software creada por Microsoft.

OpenACS: Sistema de Arquitectura Abierta de la Comunidad.

ActiveRecord: Es un patrón de diseño que se suele dar a las aplicaciones empresariales.

FTP: Protocolo de Transferencia de Archivos que permite el envío y la recepción de ficheros.

Servlets: Aplicación o pequeña aplicación Java que se ejecuta en un servidor web y que se envía al usuario junto a una página web con objeto de realizar determinadas funciones, tales como el acceso a bases de datos o la personalización de dicha páginas web.

E-business: Cualquier tipo de actividad empresarial o negocio electrónico realizada a través de las Tecnologías de la Información y las Comunicaciones.

CRUD: Es el acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Retrieve, Update y Delete en inglés).

World Wide Web Consortium: Organización que desarrolla estándares para guiar la expansión de la Web.

Kernel Linux: Núcleo de un sistema operativo, bloque de código con la parte central del funcionamiento y arranque del sistema. Centro principal del sistema operativo, encargado de hacer interactuar el software con el hardware.

Perl: Lenguaje Práctico para la Extracción e Informe, es un lenguaje de propósito general que es utilizado para un amplio rango de tareas incluyendo administración de sistemas y desarrollo web.

Python: Es un lenguaje interpretado, permite escribir programas muy compactos y legibles.

Lenguaje procedural: Es un lenguaje tradicional de programación, donde es la aplicación quien controla qué porciones de código se ejecuta, y la secuencia en que este se ejecuta.

Plugin: Pequeño programa que se adhiere a otro para poder ejecutar cierto tipo de archivos o para aportarle una función nueva, generalmente muy específica.

Ruby: Lenguaje de programación interpretado, reflexivo y orientado a objetos.

Tags: Una etiqueta o baliza (términos a veces reemplazados por el anglicismo tag) es una marca con tipo que delimita una región en los lenguajes basados en XML o HTML.

Debug: Proceso de remoción o depuración de errores en el software o en los datos. Se utiliza para referirse a código en programación.

Helpers: Programa local que se llama por un navegador web para mostrar información en un medio que puede ser texto o imágenes sencillas.

Layouts: Es la estructura de la distribución de los contenidos de un sitio web.

Lucene: Es un API de código abierto para recuperación de información, originalmente implementada en Java.

API: Interfaz de Programación de Aplicaciones (Application Programming Interface) es el conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Bibliografía citada

Achour, Mehdi, y otros. 2007. PHP Manual. 2007.

Carrion, Juan y Medina, Sergio. 2000. *Los portales y la gestión del conocimiento. Fundación iberoamericana del conocimiento.* 2000.

De Seta, Leonardo. 2009. Dos Ideas. *Introducción a los Portales Web.* [En línea] 23 de Febrero de 2009. http://www.dosideas.com/java/433-introduccion-a-los-portales-web-java.html.

Española, Real Academis. Real Academis Española. [En línea] http://www.rae.es/RAE/Noticias.nsf/Home?ReadForm.

Fernández Carrasco, Oscar M., García León, Delba y Beltrán Benavides, Alfa. 1995. Un enfoque actual sobre la calidad del software. [En línea] septiembre-diciembre de 1995. http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.

IEEE. 1990. *Standard Glossary of Software Engineering Terminology.* 1990.

Informático, Diccionario. 2009. Portal. [En línea] 2009. http://diccionario.babylon.com/Portal.

Krutchen, Philippe. Rational Software.

S: Pressman, Roger. *Ingenieria del Software un enfoque práctico.*

Woodman, Lynda. 1985. *Information management from strategies to action.* London : ASLIB, 1985.

definición. 2010. definición.org. [En línea] 2010. [Citado el: 20 de noviembre de 2009.] http://www.definicion.org/calidad-total.

Diccionario de términos de certificación. 2010. Cámara de comercio e industria de la federación rusa. [En línea] 2010. [Citado el: 18 de febrero de 2010.] http://www.gost-soex.ru/es/DICCIONARIO-DE-TERMINOS-DE-CERTIFICACION.shtml.

ISO. 2000. ISO 9000 del 2000. 2010.

Pirámide Digital. 2007. Pirámide Digital. La solución a sus problemas. [En línea] 31 de enero de 2007. [Citado el: 25 de enero de 2010.] http://www.piramidedigital.com/Tips/gerencia/diccionariogerencial.htm.

Bibliografía consultada

Alvarez, Miguel Angel. 2002. Desarrollo web. [Online] julio 8, 2002. [Cited: marzo 26, 2010.] http://www.desarrolloweb.com/articulos/831.php.

Angel Alvarez, Miguel. 2003. Desarrollo web.com. [Online] junio 4, 2003. [Cited: marzo 2, 2010.] http://www.desarrolloweb.com/articulos/1178.php.

Aptana.org. 2009. Aptana.org. [Online] 2009. [Cited: marzo 1, 2010.] http://www.aptana.org/.

De Seta, Leonardo. 2009. Dos ideas. [Online] Febrero 23, 2009. [Cited: marzo 15, 2010.] http://www.dosideas.com/noticias/java/433-introduccion-a-los-portales-web-java.html.

2010. Direccion de inforática Chubut. [Online] 2010. [Cited: enero 29, 2010.] http://www.chubut.gov.ar/informatica/docs/EstandaresCodificacion.pdf.

Fernández Carrasco, Oscar M., García León, Delba and Beltrán Benavides, Alfa. Informes Técnicos. [Online] [Cited: febrero 10, 2010.] http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm.

Fernández Guillot., Carlos and DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA. 2010. Cordinación docente de lenguajes de programación. [Online] 2010. [Cited: enero 25, 2010.] Cordinación docente de lenguajes de programación.

Hard h2o, Modding & Hardware. 2010. Hard h2o, Modding & Hardware. [Online] 2010. [Cited: mayo 4, 2010.] http://www.hard-h2o.com/diccionario-informatico_n-r.html.

Hooping.net. 2008. Hooping.net. [Online] 2008. [Cited: enero 29, 2010.] http://www.hooping.net/glossary/aplicaciones-web-146.aspx.

http://admon.8m.com. 2010. Historia. [Online] 2010. [Cited: enero 20, 2010.] http://admon.8m.com/html/edadmedia.htm.

Maestros del web. 2009. Editores web que facilitan tu trabajo. [Online] 2009. [Cited: febrero 15, 2010.] http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/.

Montero, Carlos Caballero. 2008. Aplicaciones de Escritorio, Aplicaciones Portátiles y Aplicaciones en Línea. [En línea] 2008. [Citado el: 20 de enero de 2010.] http://www.perutech.com.pe/node.php?new=104.

Multimania tripod. 2009. Multimania tripod. [Online] 2009. [Cited: marzo 5, 2010.] http://www.multimania.es/support/glossary/.

—. 2009. Multimania tripod. [Online] 2009. [Cited: abril 5, 2010.] http://www.multimania.es/support/glossary/.

Piñero, Rafael. 2009. Aplicaciones online y offline, Visual Beta. [Online] Marzo 11, 2009. [Cited: febrero 21, 2010.] http://www.visualbeta.es/8634/software-libre/aptana-studio-un-entorno-de-desarrollo-completo/.

Quiroga, Lourdes Aja. 2002. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. [Online] mayo 10, 2002. [Cited: febrero 2, 2010.] http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.

Radio santa cruz. 2005. Radio santa cruz. [Online] Mayo 29, 2005. [Cited: abril 1, 2010.] http://www.radiosantacruz.icrt.cu/pagina/glosario-informatico-a-d.htm.

Seguridad en internet es méxico. 2009. Seguridad en internet es méxico. [Online] 2009. [Cited: febrero 20, 2010.] http://www.e-indigenas.gob.mx/wb2/eMex/eMex_Glosario_de_terminos_Seguridad?page=24.

—. 2009. Seguridad en internet es méxico. [Online] 2009. [Cited: mayo 24, 2010.] http://www.eseguridad.gob.mx/wb2/eMex/eMex_Glosario_de_terminos_Seguridad?page=25.

Software guisho. 2008. Software guisho. [Online] November 24, 2008. [Cited: enero 15, 2010.] http://software.guisho.com/web-o-desktop.

Taringa. 2009. Taringa. [Online] febrero 11, 2009. [Cited: febrero 25, 2010.] http://www.taringa.net/posts/ebooks-tutoriales/2147867/Aptana-Studio-%28Uno-de-los-mejores-programas-de-programacion%29.html.

Tu función. 2007. [Online] 2007. [Cited: febrero 6, 2010.] http://www.tufuncion.com/ide-php.

Victoria. 2009. Definicion ABC. [Online] febrero 23, 2009. [Cited: febrero 27, 2010.] http://www.definicionabc.com/tecnologia/portal.php.

XMundo netwoks. 2010. XMundo netwoks. [Online] 2010. [Cited: abril 3, 2010.] http://www.xmundo.net/glosario-a.html.