Universidad de las Ciencias Informáticas

Facultad 10

Título: Sistema para la Configuración de Metodologías de Desarrollo de Software.

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

Autores: Leodán De Los Ángeles Buduén

Walfrido Serrano Pérez

Tutores: MSc. Maidely Calderón Montero

Ing. Eliurkis Díaz Terrero

Ciudad de La Habana, Mayo de 2009

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores del presente trabajo de diploma y autorizamos a la Universidad de las Ciencias Informáticas para que hagan uso pertinente de este trabajo.

Para que así conste firmamos el presente a los – días del mes de mayo de 2009.

_____ _____

Firma de Autor Firma de Autor

Walfrido Serrano Pérez Leodán De Los Ángeles Budúen

_____ _____

Firma Tutor Firma Tutor

Ing. Eliurkis Díaz Terrero MSc. Maidely Calderón Montero

AGRADECIMIENTOS

An important part of what I am is because of you, Miriam Sigas Martínez, thank you.

A mis amigos, los que saltaron la línea que separa la amistad de la hermandad, los quiero.

A mi familia, la que siempre estuvo.

A la familia en que se han convertido todos en el Dpto. Humanidades de la facultad 10, thank you Gra, gracias por estar.

A mis tutores, graciasssssssssssssss.

Leo

A mis padres y mi hermana por estar siempre ahí para mí, por ser mí inspiración y darme el apoyo para hacer este sueño realidad, gracias por confiar en mí.

A mi novia, gracias por estar, por ser tú; a su familia, por su apoyo.

A mis abuelos, mis primos, mi familia; que siempre he podido contar con ellos y los llevo conmigo a pesar de la distancia.

A mi compañero de tesis, por su amistad y por poner su mayor esfuerzo en este trabajo.

A mis amigos, por su ayuda incondicional, por poder confiar en ellos para todo.

A mis tutores, por todo su apoyo y confianza.

W@lfrido

DEDICATORIA

A Miriam Budúen Olivares, por la luz y la guía, por la vida, por lo que soy. No hay momento en que no te necesite.

Leo

A mi mamá, Melvis, por ser la mejor madre del mundo, por estar siempre en cada momento que hizo falta, por ser mi guía, mi razón de ser.

A mi papá, Walfrido, por ser el mejor padre del mundo, por poder contar contigo en cada momento, por ser mi ídolo y ejemplo a seguir.

A mi hermanita, Lisi, por ser la mejor hermana del mundo, por confiar en mí, por ser mi luz y ser tan especial para mí.

A mi novia, Yumi, por todo el amor que me has dado, por estar siempre a mi lado en las buenas y las malas, por ser el amor de mi vida.

W@lfrido

RESUMEN

El presente trabajo se refiere a la propuesta de un sistema que permite la administración de las metodologías de desarrollo de software y sus elementos correspondientes acorde con las necesidades de cada proyecto. Su desarrollo está basado en tecnologías libres, multiplataforma y sobre una arquitectura en capas utilizando PHP 5 como lenguaje de programación, se implementa el patrón de arquitectura Modelo Vista Controlador (MVC) a través del framework CodeIgniter 1.7, MySQL 5.0 como Sistema Gestor de Base de Datos y se hace uso de tecnologías AJAX (Asynchronous JavaScript And XML) para realizar más eficientes las peticiones al servidor. Da continuidad a una tesis de la cual se realiza una valoración crítica del análisis y diseño que esta abarcó y valida la propuesta de solución a través del desarrollo de un conjunto de pruebas realizadas al producto final.

ABSTRACT

The following thesis document is about the proposition of a web system to manage software development methodologies and its main components fitting to the specific needs of each project. The application was developed using free technologies, such as PHP as programming language using its framework CodeIgniter 1.7, which implements Model-View-Controller development pattern, MySQL 5 as Database Management System, it also uses AJAX to improve the way requests are made to the server. It's a continuation of a previous thesis work that proposed an analysis and design, to which a critical evaluation is carried out and finally the system is tested to validate its stability and performance.

INDICE

INTRODUCCION

Para la construcción de un software se considera medular precisar desde el mismo inicio, entre otras cuestiones, la metodología de desarrollo de software a emplear en su constitución, la cual deberá ser robusta pues así garantizará que los proyectos alcancen el éxito y se logre un producto con la calidad requerida para los clientes. En el libro El Proceso Unificado de Desarrollo de Software, Jacobson, Rumbaugh y Booch plantean que una metodología es un proceso, que aplicado al mundo del desarrollo del software se utiliza para definir Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. (1)

Las características individuales de cada proyecto han evitado la universalidad del uso de alguna en particular, de ahí la necesidad de que este sea un proceso configurable y que se ajuste al equipo de desarrollo. En la actualidad existen disímiles metodologías, que se han agrupado en dos vertientes fundamentales, las metodologías llamadas tradicionales, por citar algunas, Rational Unified Process (RUP), Microsoft Solution Framework (MSF); y las metodologías ágiles como son eXtreme Programming (XP), SCRUM, entre otras. Las primeras se enfocan en cumplir con un plan de proyecto, definido desde la fase inicial, para el que se lleva una documentación exhaustiva que requeriría para su mejor aprovechamiento una organización y disponibilidad también exhaustivas, que esté además, centralizada y debidamente organizada, lo cual constituye un problema que repercute negativamente en la dinámica del negocio puesto que no se cuenta con una herramienta donde los ingenieros de software puedan consultar, organizar y configurar los elementos fundamentales concernientes a la metodología de desarrollo que hayan seleccionado, adaptable además, a las particularidades de su proyecto, que brinde la posibilidad de hacerla extensible a otros. Este es un inconveniente común para algunas organizaciones que producen software, como la que enmarca el problema al que se le propone una solución mediante el presente trabajo.

El Centro Nacional de Tecnologías de Información (CNTI) creado en el año 2000, institución de la República Bolivariana de Venezuela que tiene como razón consolidar un sistema de tecnologías de información del Estado, que apoye la gestión de la Administración Pública, a la Comunidad Organizada y al ciudadano; y haber contribuido a la creación de una fuerte industria nacional de software, todo ello en concordancia con los principios de soberanía.

En el año 2006 se desarrolla para dicha entidad una plataforma de productividad, Red de Integración y Desarrollo (RINDE), para la producción y mantenimiento de aplicaciones informáticas, y con el fin de estimular la participación de comunidades, grupos de usuarios, organismos de la Administración Pública Nacional, instituciones, universidades, sector productivo y demás personas interesadas en el desarrollo y consolidación de la industria del Software Libre en la República Bolivariana de Venezuela.

Como una segunda versión de RINDE era preciso integrar a este una plataforma que brindara un estándar para el proceso de desarrollo de software y que pudiese ser empleada y adaptada según los requerimientos de cualquier comunidad u organización para el desarrollo de sistemas, para la cual en una investigación realizada se elaboró una propuesta del análisis y diseño de un sistema web que posibilitara la creación y configuración de metodologías de desarrollo de software con el objetivo de integrar esta herramienta a la plataforma antes mencionada.

En la actualidad no se cuenta aún con una herramienta que facilite el proceso de configuración de las metodologías de desarrollo, que pudieran emplearse en dicha entidad, persiguiendo que las diferentes entidades adscritas al CNTI hagan uso de la misma, para estandarizar el proceso de desarrollo de software, lo que constituye una situación problemática a la que el presente trabajo de diploma propone una solución que plantea como **problema científico** ¿cómo llevar a cabo el proceso de gestión y configuración de las metodologías de desarrollo de software?

Siendo el **objeto de estudio** las tecnologías para la implementación de sistemas Web.

Para lo que esta investigación presenta como **campo de acción** las tecnologías para la implementación de sistemas Web en Software Libre para el proyecto RINDE (2da fase).

Las **preguntas científicas** formuladas para el desarrollo de este trabajo, que son los lineamientos en los que se ha basado esta investigación son:

1. ¿Cuáles son los antecedentes y el estado actual de la implementación de sistemas Web?
2. ¿Cuáles son los antecedentes y el estado actual de la implementación de sistemas Web en Software Libre?
3. ¿Cómo modelar un sistema Web para darle solución al problema?

Para darle solución al problema científico formulado con anterioridad se trazó el siguiente **objetivo general:**

Implementar una aplicación web que posibilite la gestión y configuración de metodologías de desarrollo de software.

Que es sustentado por los siguientes **objetivos específicos:**

- Proponer los elementos y las tecnologías a utilizar en el desarrollo del sistema.
- Implementar una aplicación web que permita la gestión y configuración de metodologías.

Las **tareas de investigación** en correspondencia unívoca con las preguntas científicas a las que dan respuesta, enunciadas para este trabajo son:

- Sistematizar los antecedentes y el estado actual de la implementación de sistemas Web.
- Sistematizar los antecedentes y el estado actual de la implementación de sistemas Web en Software Libre.
- Modelar la propuesta de un sistema Web estándar utilizando Software Libre para el proyecto RINDE (2da Fase).

Aportes prácticos esperados del trabajo

El sistema es una herramienta web que permite la administración de las metodologías de desarrollo de software y sus elementos correspondientes, dígase fases/etapas, disciplinas, actividades, roles y artefactos; acorde con las necesidades de cada proyecto en particular.

El proceso incluye la creación, modificación, eliminación, configuración, así como la exportación de las metodologías y validación de las mismas, reportes, gráficos de relaciones entre elementos, permite listar dichos elementos y asociarlos, realizar búsquedas tanto simples como avanzadas, obtener una vista previa de la metodología que se está desarrollando y controla la gestión de usuarios, así como los roles que estos tienen dentro del sistema.

Admite la coordinación del trabajo de todo el equipo de desarrolladores a lo largo del ciclo de vida del proyecto. Provee además, los mecanismos necesarios para verificar la calidad de los entregables de un proceso de desarrollo en particular, en base a los lineamientos, prácticas o estándares establecidos.

Métodos científicos de investigación

Para el desarrollo de la investigación se utilizarán métodos teóricos y empíricos.

Métodos teóricos

- **Análisis histórico-lógico:** A través de este método se estudia la trayectoria real de los elementos que se utilizan en la implementación de sistemas Web, dígase origen y evolución de los diferentes lenguajes de programación, frameworks, Entornos de Desarrollo Integrado (en lo adelante IDE) y Sistemas Gestores de Base de Datos.

- **Análisis y Síntesis:** Para la implementación del sistema se realizó una investigación previa de los procesos que intervienen en el desarrollo de software y los principales elementos que integran las metodologías como son las fases, disciplinas, actividades, roles, artefactos y otros. Los resultados alcanzados en conjunto con el conocimiento empírico nos permitieron conseguir un dominio abarcador del tema al que proponemos solución.

Métodos empíricos

- **Entrevista:** Se realizaron múltiples entrevistas a los analistas, jefes y demás implicados del proyecto en aras de obtener información referente al funcionamiento y dinámica del negocio en el que se enmarca el problema, que ellos analizaron y cuya implementación se propone con este trabajo.

Estructuración del contenido

El presente trabajo está conformado por 3 capítulos.

En el capítulo 1 se realiza una comparación entre los lenguajes de programación para el desarrollo de aplicaciones web para concluir en las razones por las que se seleccionó PHP para la implementación del sistema, al igual que la argumentación de por qué el uso de una aplicación web y no una de escritorio. Se aborda brevemente los motivos que conllevaron a la selección del framework CodeIgniter por encima de

otros existentes como el Symfony y el Zend Framework; posteriormente se habla de los principales gestores de base de datos libres que se emplean en este tipo de aplicaciones como son MySQL y PosgreSQL y finalmente de los IDE para justificar por qué se empleó el AptanaStudio en su versión 1.1.

En el capítulo 2 se realiza una valoración crítica del diseño propuesto proveniente de una tesis que abarcó hasta el análisis y diseño del sistema, a la que se le da continuidad con el presente trabajo. También se describen las nuevas clases y operaciones necesarias para darle solución al problema.

En el capítulo 3 se refiere al diseño de las pruebas de unidad que permitan validar la solución propuesta, detallando de las mismas su objetivo, alcance y tipo, al igual que los valores utilizados para la ejecución de dichas pruebas consumando un sistema robusto que garantiza a los usuarios poder gestionar la metodología que hayan seleccionado acorde a las necesidades del proyecto a través de una correcta organización de todos sus elementos en la aplicación.

CAPÍTULO 1. Fundamentación teórica

En este capítulo se realizará una comparación entre los posibles lenguajes de programación a emplearse en el desarrollo de un sistema web como el que se propone con el presente trabajo de diploma, entre ellos ASP, Perl, Java y PHP; al igual que los frameworks, Symfony, Zend Framework y CodeIgniter; los Entornos de Desarrollo, Adobe Dreamweaver, Zend Studio, Netbeans 6.5 para PHP; los Sistemas Gestores de Base de Datos, Oracle, MS SQL Server, PostgreSQL y MySQL; y las tecnologías más utilizadas en la actualidad a nivel mundial, especificando sus características, ventajas y desventajas, concluyendo con una breve explicación del por qué de la selección consumada para el desarrollo definitivo de la aplicación, factores todos que influyen a la hora de la toma de decisiones en la implementación de cualquier sistema, lo cual es necesario en gran medida, debido a las muchas variantes que existen en todas estas ramas hoy en día.

1.1 Aplicación de Escritorio vs Aplicación Web

Con la consolidación de los sistemas web como tecnología que llegó para quedarse debido en gran medida al auge de las redes locales (empresariales, institucionales o caseras) y la popularidad de la Internet, trajo consigo la oportunidad de accederla a través de computadores y otros dispositivos móviles, lo que era algo de uso privilegiado hace algunos años, ahora es común. La Internet ha elevado y extendido aun más el concepto de aplicación Web para servir no sólo a usuarios de una pequeña red sino ubicados en cualquier sitio donde tenga acceso a la Internet. De ahí que al comenzar el desarrollo de una aplicación surge la interrogante. ¿Aplicación web o aplicación de escritorio? O lo que es lo mismo... ¿HTML o Swing/SWT/GTK/Qt...? Las ventajas y desventajas que aportan ambas filosofías de desarrollo son las causantes de que haya lugar para la duda.

Según un artículo publicado en el blog personal de Alexander Ríos quien es Consultor IT. Asesor en integración de software OSS en Colombia, se expone de las aplicaciones Web que utilizan lo que se conoce como clientes livianos (light clients) los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma. Desde el punto de vista de la arquitectura se distinguen dos lados el uno es el cliente en donde se encuentra el usuario final utilizando la aplicación por intermedio de

un navegador (Internet Explorer, Mozilla Firefox, entre otros), es aquí donde el usuario interactúa con la aplicación localizada al otro lado o servidor en donde residen realmente los datos, reglas y lógica de la aplicación.

Sin tomar en consideración las desventajas que tiene el desarrollo de una aplicación web en las que se profundizará más adelante, las ventajas que ofrece son notables, entre ellas pueden citarse:

- Instalación: la principal ventaja de un servicio web es poder acceder a él (y a los datos que contiene) desde cualquier sitio. Si el usuario accede al sistema es que tiene instalado todo lo necesario para ejecutar la aplicación: el navegador: acceso desde cualquier equipo.

- Actualizaciones. No hay que actualizar nada porque tampoco ha habido que instalar nada. Es decir la administración es nula: no tiene que instalarse, no tiene que configurarse, no se tiene que hacer nada más empezar a utilizarlo. El usuario tiene todas sus preferencias/configuraciones guardadas en el servidor web y no en su ordenador.

- Consumo de recursos para terceros: la mayor parte de consumo de ciclos de procesador, memoria, etcétera, ocurren en el servidor.

Por otra parte las desventajas también deben tomarse en consideración a la hora de tomar la decisión:

- Dependencia máxima de un tercero: en el caso del software online queda más que patente esa dependencia: si se cae la red o su servidor, no se puede contar con la aplicación.

- ¿Privacidad?: todos tus datos y tu perfil de usuario junto a todas las acciones que realizas están almacenados en equipos que pueden estar comprometidos.

Y finalmente, una interfaz de usuario rica es mucho más sencilla de programar en cualquier conjunto de herramientas gráfico (Swing/SWT/...) que en los lenguajes de programación para la Web como son el Lenguaje de Marcado de Hipertexto HTML y JavaScript. A pesar de esto la implementación de aplicaciones de escritorio trae aparejada, entre otras, las siguientes desventajas:

- Duplicidad de datos por la falta de unificación de los mismos.

- Diseminación de la información y lógica en muchas partes (cada computadora que la use).
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, controlando el acceso a los usuarios a información no relevante o privada es un caos.
- Dificultad para configurar cada una de las instalaciones dependiendo de las necesidades de cada usuario.

Añadir además, que los problemas de las aplicaciones web son controlables en su mayoría y que el panorama ha mejorado con AJAX, flash, Web 2.0, entre otras para combatir la carencia de la riqueza gráfica de las aplicaciones de escritorio que cuentan con controles inteligentes que dan mayor fluidez al trabajo del usuario. Han surgido muchos frameworks de todo tipo para facilitar las tareas: jQuery, prototype, dojo, Google Web Toolkit, cualquier implementación de JSF con componentes AJAX, etcétera. Para concluir se puede consultar una lista con 20 razones por las cuales las aplicaciones web superan a las de escritorio que se hizo pública en un artículo en internet, las mismas pueden ser consultadas en el Anexo 2. (2)

Después de haber decidido si se va a realizar una aplicación web o de escritorio es necesario tener en cuenta el paradigma de programación a utilizar para llevar a cabo la misma.

1.2 Paradigmas de Programación

Representan un enfoque particular o filosofía para la construcción del software. No existe uno mejor que otro, sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro. (3)

1.2.1 Paradigma por Procedimientos o Paradigma Imperativo

Es tal vez el más conocido y utilizado en el proceso de programación, donde los programas se desarrollan a través de procedimientos. Pascal, C y BASIC son tres de los lenguajes imperativos más importantes. La palabra latina imperare significa "dar instrucciones". El paradigma se inició al principio del año 1950 cuando los diseñadores reconocieron que las variables y los comandos o instrucciones de asignación constituían una simple pero útil abstracción del acceso a memoria y actualización del conjunto de instrucciones máquina. Debido a la estrecha relación con la arquitectura de la máquina, los lenguajes de programación imperativa pueden ser implementados muy eficientemente, al menos en principio.

El paradigma imperativo aún tiene cierto dominio en la actualidad. Una buena parte del software actual ha sido desarrollado y escrito en lenguajes imperativos. La gran mayoría de programadores profesionales son principalmente o exclusivamente programadores imperativos (Hay que añadir que los paradigmas de la programación concurrente y orientada al objeto son en realidad sub-paradigmas de la programación imperativa, así que sus adeptos también son programadores imperativos).

1.2.2 Programación Funcional

Se caracteriza por el uso de expresiones y funciones. Un programa dentro del paradigma funcional, es una función o un grupo de funciones compuestas por funciones más simples estableciéndose que una función puede llamar a otra, o el resultado de una función puede ser usado como argumento de otra función. El lenguaje por excelencia ubicado dentro de este paradigma es el LISP.

1.2.3 Paradigma Declarativo o Paradigma de Programación Lógica

Se basa en el hecho que un programa implementa una relación antes que una correspondencia. Debido a que las relaciones son más generales que las correspondencias (identificador - dirección de memoria), la programación lógica es potencialmente de más alto nivel que la programación funcional o la imperativa. El lenguaje más popular enmarcado dentro de este paradigma es el lenguaje PROLOG. El auge del paradigma declarativo se debe a que el área de la lógica formal de las matemáticas ofrece un sencillo algoritmo de resolución de problemas adecuado para usarse en un sistema de programación declarativo de propósito general.

1.2.4 Programación Orientada a Objetos

El paradigma orientado a objetos, se basa en los conceptos de objetos y clases de objetos. Un objeto es una variable equipada con un conjunto de operaciones que le pertenecen o están definidas para ellos. El paradigma orientado a objetos actualmente es el paradigma más popular y día a día los programadores, estudiantes y profesionales tratan de tomar algún curso que tenga que ver con este paradigma, podría decirse, que programar orientado a objetos está de moda.

Alrededor de 1970 David Parnas planteó el ocultamiento de la información como una solución al problema de gerenciar grandes proyectos software. Su idea fue encapsular cada variable global en un módulo con un grupo de operaciones (al igual que los procedimientos y las funciones) que permitan tener un acceso directo a la variable. Otros módulos pueden acceder a la variable sólo indirectamente, llamando a estas operaciones. Hoy se usa el término objeto para tales módulos o variables encapsuladas a sí mismas. Lenguajes imperativos como Pascal y C han sido modificados (o añadidos) para que soporten el paradigma orientado a objetos para dar Delphi en el caso de Pascal y C++ en el caso de C.

Una de las bondades importantes de los lenguajes orientados a objetos es que las definiciones de los objetos pueden usarse una y otra vez para construir múltiples objetos con las mismas propiedades o modificarse para construir nuevos objetos con propiedades similares pero no exactamente iguales. El lenguaje orientado a objetos por excelencia es Smaltalk[1] desarrollado en Palo Alto Research Center durante los 1970's.

¿Pero qué es exactamente un lenguaje orientado a objetos? Los siguientes conceptos señalan las características generalmente aceptadas acerca de los lenguajes orientados a objetos.

- Objetos y clases son obviamente los conceptos fundamentales. Una clase es un conjunto de objetos que comparten las mismas operaciones.

[1] Primer sistema puro de objetos. Todo en Smalltalk es un objeto y toda la computación es desarrollada mediante mensajes que son enviados entre los objetos.

- Objetos (o al menos referencia a objetos) deben ser valores de la clase base. Así, cualquier operación puede tomar un objeto como un argumento y puede devolver un objeto como resultado. De esta manera el concepto de clase de objetos está relacionado con el concepto de tipo de dato.
- Herencia es también vista como un concepto clave dentro del mundo de los objetos. En este contexto, la herencia es la habilidad para organizar las clases de objetos en una jerarquía de subclases y superclases y las operaciones dadas para una clase se pueden aplicar a los objetos de la subclase.

Una vez analizado los distintos paradigmas de programación y ver el más apropiado para dar solución al problema planteado, se debe hacer un estudio del lenguaje de programación a utilizar, el cual depende en gran medida de la selección anterior.

1.3 Lenguaje de programación

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. (4)

1.3.1 ASP

Microsoft Active Server Page (ASP) introducida en 1996, es una tecnología de script que corre del lado de servidor y puede ser usado para crear aplicaciones Web dinámicas e interactivas. Una página ASP es una página HTML que contienen scripts que corren del lado del servidor que son procesados por un servidor Web antes de ser utilizado por el navegador. Usted puede combinar ASP con XML (Extensible Markup Language) para crear sitios Web interactivos poderosos.

ASP es una característica de Microsoft Internet Information Server. Debido a que los scripts que corren en servidor son construidos en una página regular de HTML, este puede ser servido en casi cualquier navegador. Un archivo ASP puede ser creado incluyendo un script escrito en VBScript o JavaScript en un archivo HTML. (5)

1.3.2 Java

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90 con el que podemos realizar cualquier tipo de programa.

Los programas en Java pueden ejecutarse en cualquiera de las siguientes plataformas, sin necesidad de hacer cambios: (6)

- Windows 95 y NT
- Power/Mac
- Unix (Solaris, Silicon Graphics, entre otros)

La compatibilidad es total:

- A nivel de fuentes: El lenguaje es exactamente el mismo en todas las plataformas.
- A nivel de bibliotecas: En todas las plataformas están presentes las mismas bibliotecas estándares.
- A nivel del código compilado: el código intermedio que genera el compilador es el mismo para todas las plataformas. Lo que cambia es el intérprete del código intermedio.

Desventajas

Dado que la máquina virtual de java es un intérprete y redunda en una falta de rendimiento con relación a aplicaciones equivalentes escritas en código máquina nativo. Una respuesta a este problema es el empleo de compiladores JIT (Just In Time). Un compilador JIT interacciona con la máquina virtual de java para convertir el código de bytes en código nativo.

El poder reducir los problemas de acceso a memoria y liberación automática hacen de java un lenguaje poco apropiado para desarrollar aplicaciones de base como Sistemas Operativos. (7)

1.3.3 Perl

Es un lenguaje interpretado sencillo, flexible, poderoso y eficiente. Fue creado por Larry Wall en 1987, y desde entonces ha sido mejorado y ampliado por desarrolladores de todo el mundo, convirtiéndose en una imprescindible herramienta para el programador y administrador de sistemas.

Puede utilizarse tanto para la creación de programas gráficos, como para el armado de páginas dinámicas en la web, la automatización de tareas y muchas otras aplicaciones. (8)

Algunas de las ventajas del uso del lenguaje Perl son las siguientes: (9)

- Construcción de pequeños programas que pueden ser usados como filtros para obtener información de ficheros, realizar búsquedas, etc.
- Se puede utilizar en varios entornos, como puede ser Windows, Macintosh, Unix, sin realizar cambios de código, siendo únicamente necesario la introducción del intérprete PERL correspondiente a cada sistema operativo.
- También es uno de los lenguajes más utilizados en la programación de CGI scripts, que son guiones o scripts que utilizan el interface CGI (Common Gateway Interface), para intercambio de información entre aplicaciones externas y servicios de información. Como ejemplo de ello tenemos los programas de búsqueda usados por el browser Netscape.
- El mantenimiento y depuración de un programa en PERL es mucho más sencillo que la de cualquier programa en C.

1.3.4 PHP

Fue concebido en el otoño de 1994 por Rasmus Lerdorf. Se conoce originalmente como Personal Home Pages y su primera versión salió en los comienzos de 1995. Es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Dentro de las operaciones que pueden realizarse con este lenguaje se encuentran: procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies, entre otras. (10)

Existen tres campos en los que se usan scripts escritos en PHP.

- Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. El intérprete PHP (CGI ó módulo), un servidor web y un navegador.
- Scripts en la línea de comandos. Puede crear un script PHP y correrlo sin ningún servidor web o navegador.
- Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si conoce bien PHP, y quisiera utilizar algunas características avanzadas en programas clientes, puede utilizar PHP-GTK para escribir dichos programas.

Características

- Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS, entre otros.
- Soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros.
- Aunque no todas las características estándar de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas bibliotecas y aplicaciones grandes (incluyendo la biblioteca PEAR) están escritas íntegramente usando programación orientada a objetos.
- No se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash. También puede presentar otros resultados, como XHTM y archivos XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

- La característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos, entre ellas: mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 y OCI8), Ovrimos, PostgreSQL, y muchas más.

La potencialidad de este lenguaje sobre los antes mencionados, al menos en el desarrollo de aplicaciones Web, es bien evidente, ya sea por ser "código abierto", multiplataforma o por el gran soporte de bases de datos con que cuenta. Por esto la selección de este para la implementación del sistema que defiende este trabajo de diploma.

La gran facilidad y ahorro de tiempo y código, que actualmente brindan los frameworks, es sin dudas un aspecto a considerar para llevar a cabo el desarrollo de una aplicación web y la utilización de estos está dado por el lenguaje de programación seleccionado, al ser PHP el lenguaje a utilizar para la implementación del sistema que defiende este Trabajo de Diploma, se muestra a continuación el estudio realizado sobre los mismos.

1.4 Frameworks PHP

Se define un framework, resumiendo el concepto de varias fuentes consultadas, como un conjunto de herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera generalista; es, en el ámbito informático, la estructura básica de una base de datos, proceso o programa. Su utilización simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, el empleo de un framework facilita la programación de aplicaciones, ya que encapsulan operaciones complejas en instrucciones sencillas.

Aunque los frameworks existen desde hace décadas, y los frameworks para aplicaciones web también se utilizan desde hace mucho tiempo, el lanzamiento de Ruby On Rails[2] en 2004 supuso una revolución en el

[2] Entorno de programación (Rails) que se apoya en el lenguaje Ruby. Goza de gran popularidad para el desarrollo de aplicaciones de tipo Web 2.0.

desarrollo de las aplicaciones web que aún hoy continúa dando origen a la gran diversidad de frameworks que en la actualidad existen.

1.4.1 Symfony

Primera versión publicada en Octubre de 2005 por Fabien Potencier.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web, la lógica de negocio, la lógica de servidor y la presentación. 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 PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel, entre los que se cuentan Yahoo Bookmarks (20 millones de usuarios y 12 idiomas) y partes de Yahoo Answers y del.icio.us[3]. 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 *nix (Unix, Linux, etc.) como en plataformas Windows.

La primera versión de Symfony fue publicada en Octubre de 2005 por Fabien Potencier, fundador del proyecto. Fabien es el presidente de Sensio (http://www.sensio.com/), una empresa francesa de desarrollo de aplicaciones web conocida por sus innovaciones en este campo.

Se puede citar como desventajas que, para su utilización se requiere de la generación de grandes cantidades de código y una configuración exhaustiva de antemano. También que precisa de una estructuración estricta de sus directorios que se establece por consola a la hora de crear módulos.

[3] Un servicio de gestión de marcadores sociales en la web.

Finalmente señalar, como dato curioso, que el nombre de este framework proviene del hecho que su fundador quería un nombre corto, que tuviera una ‘s’ por Sensio, el nombre de la compañía y una ‘f’ por framework, que además fuera fácil de recordar y que no estuviera asociado al nombre de ninguna otra compañía, Symfony, que aunque no es una palabra correcta en inglés (symphony), está acorde con lo que él estaba buscando.

1.4.2 Zend Framework

Concebido a principios del 2005 por la compañía Zend Technologies.

Zend Framework es un framework de código abierto para el desarrollo de aplicaciones y servicios web con PHP 5. Entre sus principales características resaltan su total y pulida orientación a objetos. Presenta además, una interdependencia baja entre sus diferentes componentes, lo que permite la utilización individual de estos.

A pesar de que pueden emplearse por separados si se integran sus componentes constituyen un robusto y extensible framework para el desarrollo de aplicaciones que ofrece una muy completa implementación del patrón de diseño Modelo-Vista-Controlador, en lo sucesivo MVC; una capa de abstracción de datos fácil de utilizar y un conjunto de utilidades para agilizar el trabajo con componentes HTML como formularios, validaciones, entre otros.

Otros componentes tales como el Zend_Auth y el Zend_Acl proveen ayuda para la autenticación de usuarios contra la mayoría de los proveedores de credenciales existentes. Cabe mencionar también el Zend_Config para temas de configuración de aplicaciones web, Zend_Db para tratar con bases de datos, Zend_Search o Zend_Feed entre otros como para la gestión de documentos en PDF, canales RSS, etcétera.

1.4.3 CodeIgniter

Desarrollado por Rick Ellis para EllisLab, Inc. Primera versión lanzada 28 de febrero de 2006.

El creador de PHP Rasmus Lerdof en la conferencia frOSCon de agosto del 2008, haciendo alusión al rendimiento de los frameworks de PHP existentes en la actualidad acotó que le gustaba de entre todos CodeIgniter porque es más rápido, ligero y el que menos se parece a un framework.

De la traducción del manual oficial del framework se tiene que CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada.

CodeIgniter se encuentra bajo licencia de código abierto Apache/BSD.

Características

- Sistema basado en Modelo-Vista-Controlador.
- Compatible con PHP 4.
- Extremadamente liviano.
- Clases de base de datos llenas de características con soporte para varias plataformas.
- Soporte de Active Record para Base de Datos
- Formulario y validación de datos.
- Seguridad y filtro XSS
- Manejo de sesión
- Clase de envío de correo. soporta archivos adjuntos, correo de texto/HTML, múltiples protocolos (sendmail, SMTP, Mail) y más.
- Librería de manipulación de imagen (cortar, redimensionar, rotar, etc.), soporta GD, ImageMagick, y NetPBM

- Clase de carga (upload) de archivo
- Clase de FTP
- Localización
- Paginación
- Encriptación de datos
- Puntos de referencia
- Cacheo de páginas enteras
- Historial de errores
- Perfilando la aplicación
- Clase de calendario
- Clase de agente del usuario
- Clase de codificación Zip
- Clase de Motor de Plantillas
- Librería XML-RPC
- Clase de prueba de unidad
- URLs amigables a motores de búsqueda
- Ruteo de URI flexible
- Soporte para ganchos, extensiones de clase y plugins
- Larga librería de funciones "asistentes"

Ventajas por las cuales utilizar CodeIgniter

- Es un framework verdaderamente liviano.
- Es un framework que requiere apenas de configuraciones.
- Es un framework que no precisa el uso de herramientas de línea de comando (consola).
- Es un framework para cuyo uso no se requiere seguir estrictas reglas de codificación.
- Se utiliza cuando no se necesita de librerías de gran escala como el Repositorio de Extensiones y aplicaciones de PHP (PEAR[4]).
- No se precisa ser forzado a aprender un lenguaje de plantilla para su utilización (aunque dispone de uno si se quiere utilizar).
- Se evita la complejidad, arribando a soluciones más simples.
- Se dispone de documentación a lo largo de todo el proceso.

Luego de haber determinado el framework, los paradigmas y el lenguaje de programación a emplear, se hace necesario un Entorno de Desarrollo para comenzar con la implementación del sistema.

1.5 Entornos de Desarrollo para PHP

Los IDEs son un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un buen editor de código, administrador de proyectos y archivos, enlace transparente a compiladores y depuradores e integración con sistemas controladores de versiones o repositorios. (11)

[4] Es un framework y sistema de distribución de compones reutilizables de PHP.

1.5.1 Zend Studio

Es un completo Entorno de Desarrollo Integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. Además, sirve de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. (12)

Algunas de las características que posee son:

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- PhpDoc integrado.
- Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- Inserción automática de paréntesis y corchetes de cierre.
- Sangrado automático y otras ayudas de formato de código.
- Emparejamiento de paréntesis y corchetes.
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform).
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.

- Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador).
- Cliente FTP integrado.
- Soporte para navegación en bases de datos y ejecución de consultas del Lenguaje de Consultas Estructurado SQL (Structured Query Language).

1.5.2 Adobe Dreamweaver

Programa desarrollado por Macromedia (ahora Adobe) para diseño, desarrollo, publicación y administración de sitios y aplicaciones Web, además de ser reconocido por la buena gestión con los archivos CSS (13)

Además de sus capacidades WYSIWYG[5], tiene las funciones típicas de un editor de código fuente para la web:

- Un administrador de sitios, para agrupar los archivos según el proyecto al que pertenezcan.
- Un cliente FTP integrado, que permite subir los archivos editados inmediatamente al sitio en Internet.
- Función de autocompletar y resaltado de la sintaxis para instrucciones en HTML y lenguajes de programación como PHP, JSP o ASP.

1.5.3 Netbeans 6.5 para PHP

NetBeans es una aplicación de código abierto ("open source") para desarrolladores. Posee todas las herramientas necesarias para crear aplicaciones web, de escritorio y para teléfonos móviles con los lenguajes de programación Java, C, C++ e incluso lenguajes dinámicos como: PHP, JavaScript, Groovy, y Ruby. Es fácil de usar, instalar y puede ser ejecutado en múltiples plataformas como Windows, Linux, Mac OS X y Solaris. (14)

[5] Siglas en inglés de ***(****What You See is, What You Get****):*** Se utiliza para indicar que lo que se ve en pantalla, es como se vería de forma impresa. (15)

Entre sus principales características podemos mencionar:

- Brinda completamiento automático de código PHP, así como coloreado de código sintáctico y semántico.
- Permite depurar el código usando Xdebug.
- Genera fragmentos de código para bases de datos MySQL.

Dadas las características, ventajas y soporte que brinda este potente Entorno de Desarrollo y sobre todo por ser su licencia "open source", se eligió como IDE a utilizar para el desarrollo del sistema, además de que los antes mencionados (Zend Studio y Dreamweaver), tienen similares características pero son softwares privativos, estando en contra de la política seguida para el desarrollo de dicha aplicación.

Para la continua manipulación de datos que trae consigo una aplicación, es preciso recurrir a un gestor de base de datos que se encargue de contener la información necesaria para el posterior uso de esta.

1.6 Gestores de Base de Datos

Sintetizando una serie de conceptos consultados, se define una Base de Datos como un conjunto de datos almacenados con una estructura lógica compartidos por varios procesos de forma simultánea para un propósito específico, que pueden ser consultados y manipulados. Es decir, tan importante como los datos, es la estructura conceptual con la que se relacionan entre ellos. En la práctica debe añadirse a esta definición los programas (o software) que hacen de ellos un conjunto consistente puesto que la combinación de ambos da la coherencia requerida para poder trabajar con los datos de una manera sistemática, a lo que se denomina Sistema Gestor de Base de Datos (SGBD) que no es más que un sistema software que facilita la creación, el mantenimiento y uso de una base de datos electrónica, según el diccionario Wordnet de la Universidad de Princeton, en otros conceptos similares se tiene que es una colección de programas que se utiliza para almacenar, modificar y extraer información de una base de datos (www.studiodog.com/glossary-d.html). De estos conceptos se deduce que un SGBD facilita las funciones de:

- Almacenar físicamente

- Garantizar consistencia
- Garantizar integridad
- Atomicidad transaccional
- Manejo de vistas a la información

Ejemplos de estos existen varios, propietarios y de software libre, entre los que pueden nombrarse, entre los primeros Oracle, MS SQL Server, Informix, entre otros, y entre los gestores libres, PostgreSQL, MySQL, etcétera.

1.6.1 Oracle

(Última versión: 11g Release 1 (2007))
SGBD de Oracle Corporation (http://www.oracle.com), creado a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre los SGBD de George Koch, es a todas luces el líder entre los SGBD a nivel empresarial para el comercio electrónico. La base de datos Oracle es ampliamente utilizada en diversos tipos de grandes aplicaciones, y es mayormente popular porque las características que la definen apenas dejan ver desventajas:

- Versatilidad: la Corporación Oracle ofrece muchos productos de comercio electrónico que integran con su base de datos, lo que permite acelerar el proceso de diseño y construcción de la aplicación que utiliza esta base de datos.
- Estabilidad: los administradores reportan que los servidores de base de datos de Oracle rara vez fallan, óptimo si se tiene una aplicación que requiere un funcionamiento a tiempo completo.
- Disponibilidad de Interfaz de Usuario gráfica (en lo adelante GUI): Oracle ofrece muchas herramientas con GUI para gestionar el servidor de base de datos.
- Seguridad: las versiones actuales de Oracle incluyen un conjunto de herramientas que permite la encriptación de datos sensibles dentro de la base de datos. También provee, al igual que otros

gestores, seguridad a nivel de usuario para proteger la información de ataques de usuarios malintencionados o errores de los administradores.

- Soporte y mantenimiento: la compañía Oracle destaca por históricamente responder a las solicitudes de usuarios de integrar nuevas características, también responde a las tendencias del mercado, manteniendo siempre la documentación disponible a su red de usuarios.

- Multiplataforma: versiones populares incluyen Oracle tanto para Windows como para Linux, aun para este último, las versiones son privativas.

Puede acotarse como desventaja que es un SGBD que para su correcto funcionamiento requiere del consumo de altos recursos de hardware, dígase velocidad del procesador y capacidad de RAM.

1.6.2 Microsoft SQL Server

(Última versión 2008 (10.00.1600.22) (2 de Agosto de 2008))
A semejanza de Oracle, Microsoft constituye un jugador clave en el mercado de las base de datos, campo en el que se inserta con el SGBD MS SQL Server, es basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea, con las características que han hecho de Microsoft el líder incuestionable de los sistemas operativos de escritorio:

- Estabilidad bastante alta: MS SQL Server ofrece un nivel de estabilidad diseñado para estar acorde con la de su sistema operativo Windows. Razón por la cual algunos inversores son renuentes a invertir en MS SQL Server como una solución para sus necesidades de una base de datos, por los conocidos problemas de seguridad de Windows

- Facilidad de uso: MS SQL Server funciona con una interfaz de usuario al estilo de Windows, lo que permite acelerar la curva de aprendizaje.

- Regido por el estándar ANSI SQL-92: MS SQL Server se apega completamente al estándar, además lo extiende, al igual que otros SGBD

- Disponibilidad de Soporte: Directamente desde Microsoft e incluso otros vendedores, existe una buena documentación y respuesta a las solicitudes de los usuarios.

Como desventajas cabe mencionar que al igual que el sistema operativo en el que corre, MS SQL Server consume altos recursos de hardware, nos es multiplataforma, por consiguiente crea dependencia del vendedor, para actualizar el servidor o el software de la base de datos se requiere reiniciar el ordenador donde está instalado, lo cual es inaceptable para aquella aplicaciones que necesiten proveer una disponibilidad absoluta.

1.6.3 PostgreSQL

(Última versión 8.3 4 de febrero de 2008))
PostgreSQL es un SGBD objeto-relacional basado en POSTGRES, versión 4.21, desarrollado en la Universidad de California en el Departamento de Ciencias de la Computación de Berkeley. Se encuentra publicado bajo la licencia libre BSD.

Postgres fue pionero de muchos conceptos que solo tiempo después fueron incorporados a algunos SGBD comerciales. PostgreSQL es un descendiente de código abierto del original de Berkeley. Soporta una gran parte de los estándares de SQL y ofrece muchas características modernas que aportan potencia y flexibilidad adicional, entre ellas:

- Objeto-relacional: en PostgreSQL cada tabla define una clase. PostgreSQL implementa la herencia entre tablas y los operadores y las funciones son polimórficos.
- Consultas complejas.
- Claves primarias y foráneas (Integridad referencial).
- Disparadores.
- Vistas.
- Integridad transaccional.
- Control de concurrencia multiversión.

- Reglas.
- Restricciones.
- Múltiples lenguajes procesales: Los disparadores y otros procedimientos pueden escribirse en cualquiera de los varios lenguajes procesales. El código del lado del servidor normalmente está escrito en PL/pgSQL, pero puede también desarrollarse en TCL, Perl e incluso Bash en entornos Linux.
- Múltiples API[6]s (**A**pplication **P**rogramming **I**nterface) del lado del cliente: PostgreSQL soporta el desarrollo de aplicaciones clientes en varios lenguajes de programación, entre ellos C, C++, ODBC, Perl, PHP, Tcl/Tk, y Python.

Además es extensible a las necesidades de los usuarios que pueden añadir nuevos:

- Tipos de datos.
- Funciones.
- Operadores.
- Funciones agregadas.
- Lenguajes procesales.

Válido mencionar también como características que potencian al SGBD PostgreSQL que es multiplataforma, se rige por el estándar SQL92 e incluye algunas características del estándar SQL99, en comparación con otros SGBD consume bajos recursos de hardware. Como desventaja que ha frenado un tanto su adopción a gran escala está que es más lento que algunos de sus competidores como Oracle en el manejo de datos transaccionales.

[6] Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

1.6.4 MySQL

(Última versión 5.1.30 (27 de Noviembre de 2008))
MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad. (16)

MySQL es un SGBD relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Interioridades y portabilidad

- Escrito en C y en C++
- Multiplataforma.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-hilo mediante hilos del kernel. Pueden usarse fácilmente múltiples CPUs si están disponibles.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Un sistema de reserva de memoria muy rápido basado en hilos.
- Combinaciones de registros de dos o más tablas (joins) muy rápidas usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.

- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

Seguridad

Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor; determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.

Escalabilidad y límites

- Soporte a grandes bases de datos. Se emplea MySQL Server con bases de datos que contienen 50 millones de registros.

- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

MySQL se ha convertido en el SGBD de código abierto más popular del mundo debido a su consistentemente rápido rendimiento, alta confiabilidad y facilidad de utilización. Es utilizado tanto por desarrolladores individuales como por grandes compañías líderes entre las que se incluyen Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube y Zappos.com.

Su empleo ha sido potenciado por su inclusión en el paquete LAMP (Linux, Apache, MySQL, PHP / Perl / Python.) para el desarrollo de una nueva generación de aplicaciones. MySQL corre en más de 20 plataformas incluyendo GNU/Linux, Windows, OS/X, HP-UX, AIX, Netware.

MySQL ofrece un extenso rango de software certificado, soporte, entrenamiento y consulta para garantizar el éxito de cualquier producto.

Sintetizando se listan a continuación, de forma concreta, algunas ventajas y desventajas a tomar en cuenta a la hora de decidirse por este SGBD para la implementación del sistema.

Ventajas

- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- Conectividad: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o sockets Unix.
- Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador, lo que repercute directamente en una de sus principales ventajas: mayor rendimiento.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable, sin límites en los tamaños de los registros

Influyen también en la decisión las desventajas que posee este SGBD, entre ellas:

Desventajas

- Integridad referencial: aunque admite la declaración de claves ajenas en la creación de tablas, internamente no las trata de forma diferente al resto de los campos.
- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

Después del estudio realizado al respecto de las diferentes variantes actuales en el desarrollo de aplicaciones web, dígase lenguajes de programación, frameworks, sistemas gestores de base de datos, entornos de desarrollo, se concluye que para llevar a cabo la implementación del sistema, se empleará el lenguaje de programación PHP 5 en su versión 5.2, con el framework CodeIgniter versión 1.7.1, sobre el entorno de desarrollo Netbeans 6.5 para PHP y utilizando como SGBD MySQL en su versión 5.0, por las

facilidades que proveen en conjunto vistas con anterioridad en el cuerpo del capítulo. Todas las herramientas utilizadas tienen en común que son estandartes del software libre, líderes en su mayoría en este ámbito.

CAPÍTULO 2. Descripción y análisis de la solución propuesta.

El presente capítulo se centra en la implementación del sistema al que se le propone solución con este trabajo de diploma, para ello, se realizará un valoración crítica del diseño propuesto por el analista, donde se estudiarán los cambios necesarios para la transición del diseño a la implementación, teniendo en cuenta los requerimientos de la aplicación. Se abordarán cuestiones referentes a los componentes a reutilizar, dígase clases o librerías previamente implementadas, al igual que las estrategias de integración. Posteriormente se describirán los algoritmos no triviales a implementar, realizando un análisis de la complejidad de los mismos, para finalmente describir las nuevas clases y operaciones necesarias para darle solución a la situación problemática.

2.1 Valoración crítica del diseño propuesto por el analista.

A continuación se muestran los diagramas de clases del diseño propuestos por el analista y la versión final después de realizarle los cambios acorde a los requerimientos de la aplicación durante el flujo de trabajo de implementación. Los cambios obedecen fundamentalmente a la utilización del framework para PHP CodeIgniter, que utiliza el patrón arquitectónico MVC, cuyo funcionamiento se explica en detalle en el siguiente epígrafe y que repercute en la estructuración de las clases para adecuarse al patrón aplicado.

Pueden citarse entre los principales cambios a la propuesta del analista, que las clases cliente son construidas por una clase en el servidor que contiene la clase controladora del módulo, que es la que tiene funciones que generan las vistas (clases cliente) y dichas vistas envían la información a la misma clase en el servidor, donde en la propia función de la controladora se gestiona el flujo de la información hacia una clase librería que media entre la controladora y la modelo, esta clase librería es incluida por la controladora permitiendo un mayor desempeño porque brinda un conjunto de utilidades para facilitar el flujo de datos entre la controladora y la modelo. Además de los cambios generales que se aplicaron a la propuesta del analista, los paquetes sufrieron algunos cambios más específicos, que se enuncian a continuación.

Diagrama de Clases del Diseño (DCD) (Paquete Metodología)

Se añadieron las funcionalidades para:

- Habilitar metodología
  - Se añadió la página cliente con un formulario que contiene un campo Id_Metodología para entrar un número entero que representa la metodología que se desea habilitar.
- Deshabilitar metodología
  - Se añadió la página cliente con un formulario que contiene un campo Id_Metodología para entrar un número entero que representa la metodología que se desea deshabilitar.

Algunos formularios fueron sujetos a cambios según peticiones del cliente, ejemplo de ello, el de la clase cliente cp_Crear Metodología, se le quitó el campo Iteraciones, y se le añadieron los campos Usuario con permiso, que permite seleccionar cuáles usuarios tendrán permisos para editar la metodología que se está creando; y el campo Principios para entrar una cadena de caracteres, que especificarán los principios por los que se regirá la metodología. También se analiza la propuesta para los módulos Fase, Disciplina, Actividad, Rol y Artefacto.

Los módulos Fase, Disciplina, Actividad, Rol y Artefacto implementan el patrón de casos de uso CRUD (Create, Read, Update, Delete) que por sus siglas en inglés se refiere al conjunto de operaciones básicas en bases de datos o la capa de persistencia en un sistema de software que permiten crear, leer o consultar, actualizar y borrar. De ahí que en el presente epígrafe, se toma solamente un módulo para la valoración crítica, debido a que los demás están sujetos a transformaciones similares y de esta forma fueron modelados por el analista.

DIAGRAMA DE CLASES (Paquete Metodología) propuesto por el analista

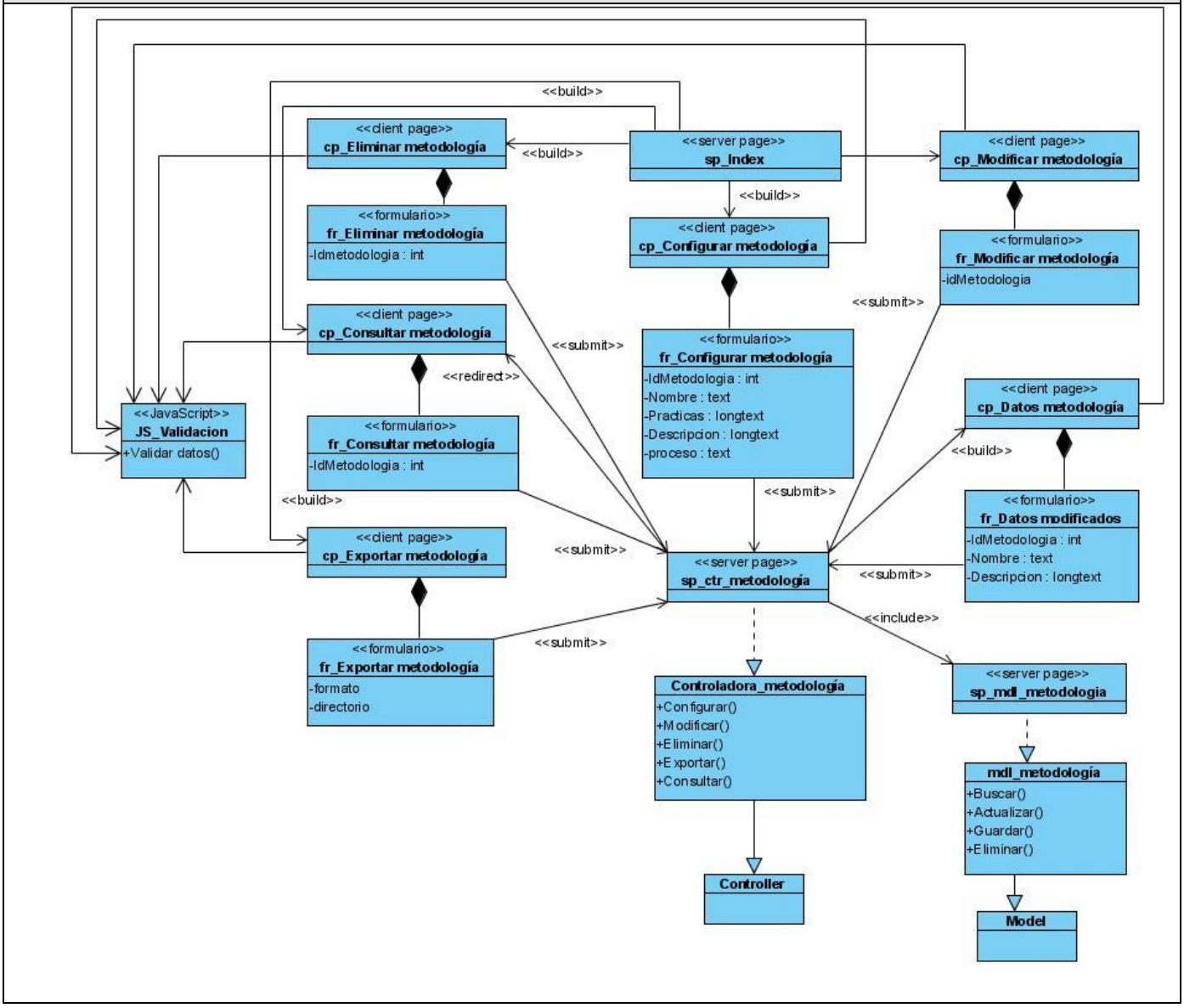


DIAGRAMA DE CLASES (Paquete Metodología) versión final

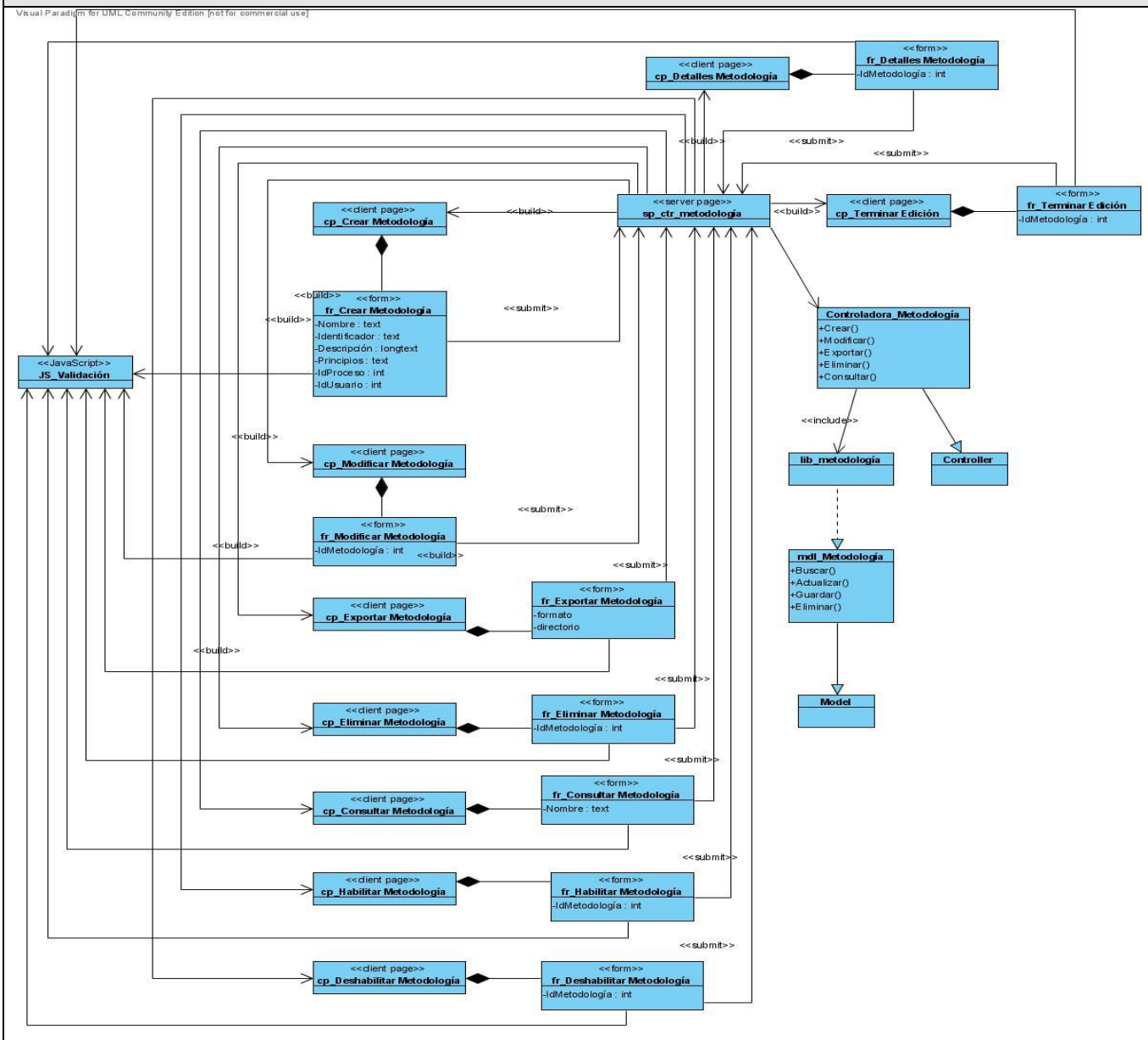


DIAGRAMA DE CLASES (Paquete Fase) propuesto por el analista

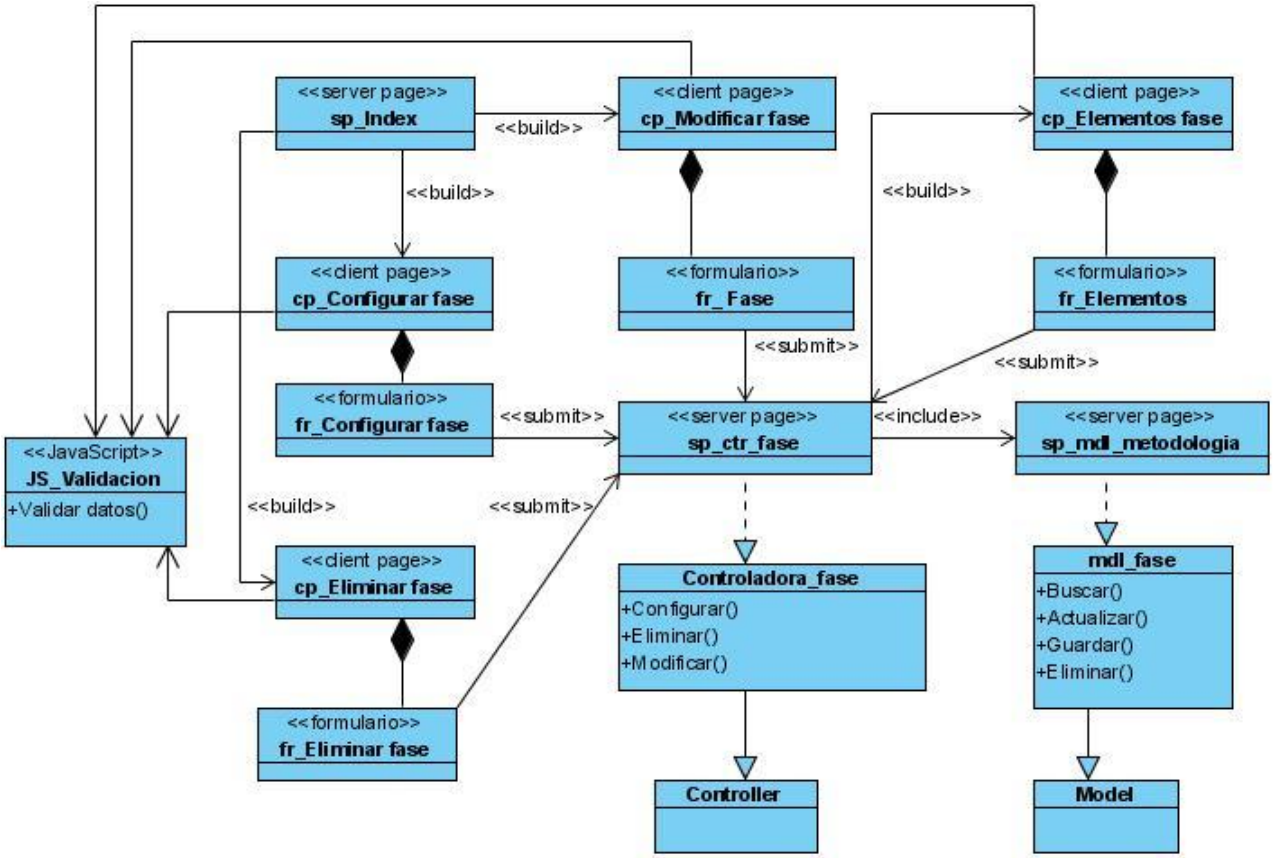
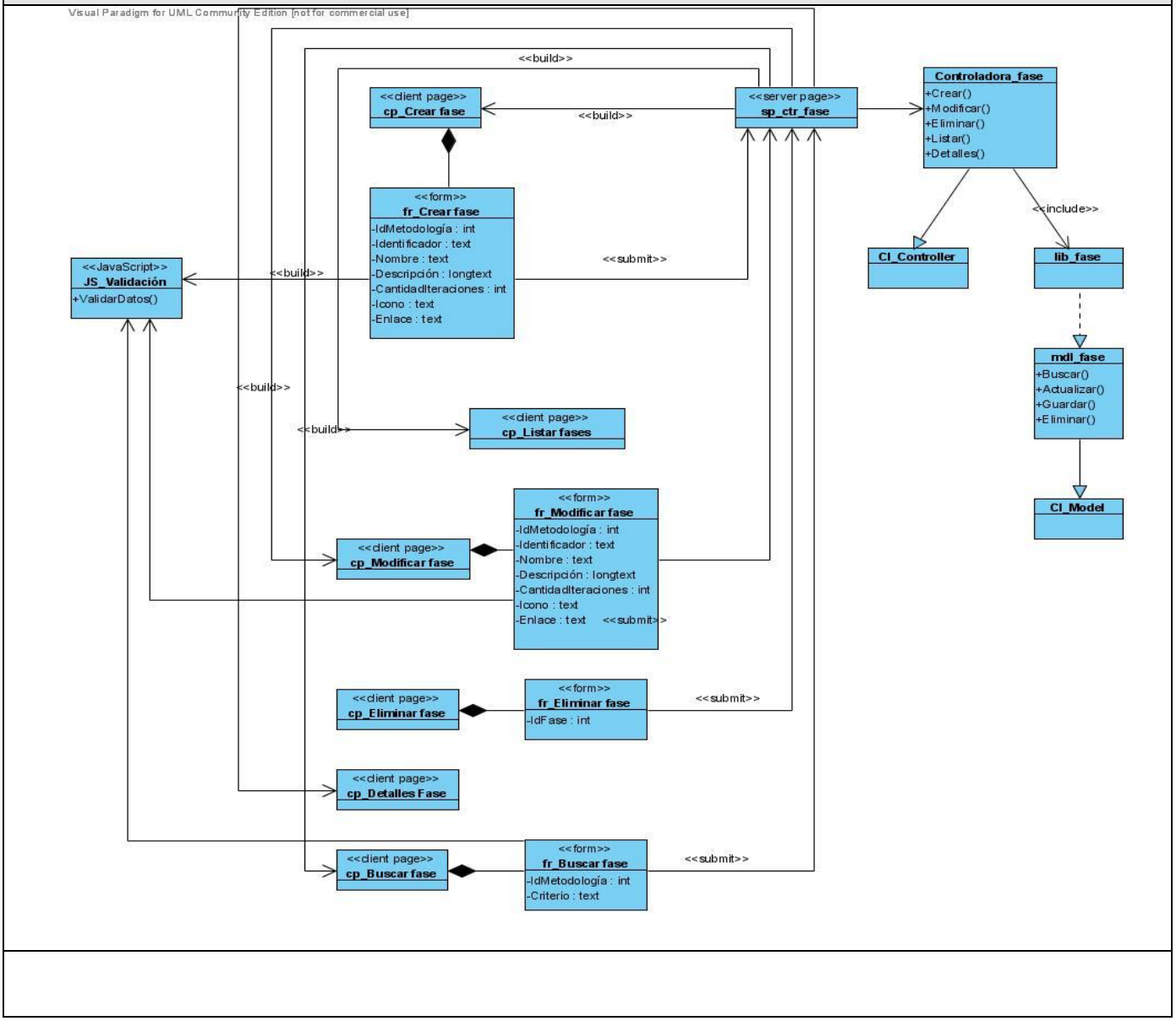


DIAGRAMA DE CLASES (Paquete Fase) versión final



Posterior a la valoración crítica de la propuesta de diseño del analista, que tiene una gran repercusión puesto que decide qué se asume y qué se modifica para la implementación del sistema, debe analizarse de igual manera la reutilización de componentes o módulos ya existentes, que pueden ser reutilizados, así

como las estrategias de integración, el resultado de este análisis se traduce en un avance considerable en la implementación por concepto de ahorro de tiempo y esfuerzo.

2.2 Análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados. Estrategias de integración.

CodeIgniter

En el capítulo anterior se expuso que por sus características y potencialidades, se decidió emplear, para la implementación del sistema, el framework de PHP CodeIgniter. Entre las características que lo describen se mencionó que implementa el patrón arquitectónico Modelo-Vista-Controlador (MVC) que permite una buena separación entre lógica y presentación. A continuación se ilustra el funcionamiento de dicho patrón a través de un gráfico:

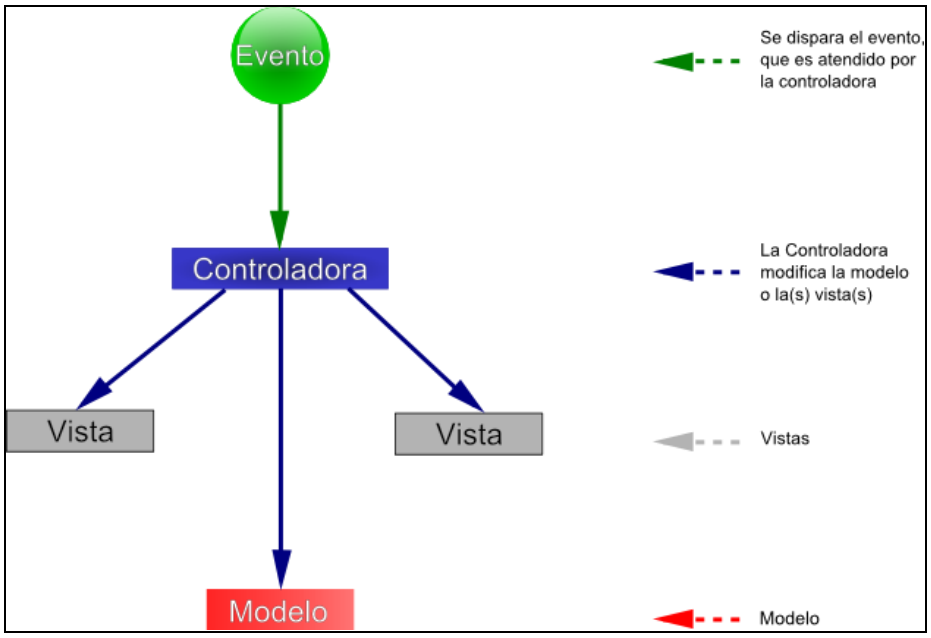


Figura 1. Implementación del patrón Modelo Vista Controlador según el framework CodeIgniter.

MVC es una aproximación al software que separa la lógica de la aplicación de la presentación. En la práctica, permite que las páginas web contengan mínima codificación ya que la presentación es separada del código PHP.

El Modelo representa la estructura de datos. Típicamente las clases de modelo contendrán funciones que para recuperar, insertar y actualizar información en la base de datos.

La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un píe de página. También puede ser una página RSS, o cualquier otro tipo de "página".

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web.

CodeIgniter tiene un enfoque bastante flexible del MVC, ya que los Modelos no son requeridos. Si no se necesita agregar separación, o mantener los modelos requiere mayor complejidad, pueden ser ignorados y construir la aplicación mínimamente usando Controladores y Vista.

CodeIgniter también permite incorporar códigos existentes, o incluso desarrollar librerías de núcleo para el sistema.

Cabe mencionar entonces cómo fluyen los datos a través del sistema:

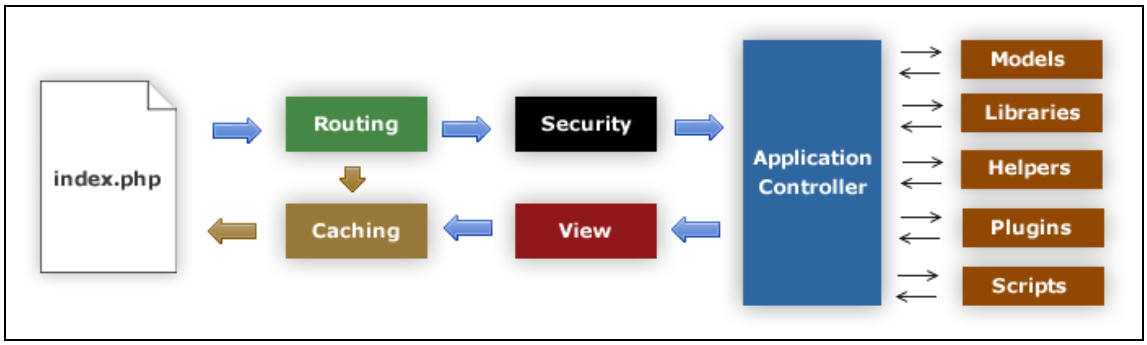


Figura 2. Diagrama de Flujo de la Aplicación

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
2. El Router examina la petición HTTP para determinar que debe ser hecho con él.
3. Si un archivo de caché existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.

4. Seguridad. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
5. El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.
6. La Vista finalizada es presentada entonces enviada al explorador web para ser vista. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas.

Otras librerías que se investigaron para su reutilización son FPDF, HTML2PDF, PHPMailer, Ldap, Template, Message, Json, que brindan un conjunto de funcionalidades para la gestión de documentos en PDF (Portable Document File), el envío de correo electrónico, la comunicación mediante el protocolo LDAP (Lightweight Directory Access Protocol), la creación de plantillas y la gestión de mensajes para CodeIgniter, entre otras.

FPDF

Fpdf es una biblioteca escrita en lenguaje de programación PHP que permite crear archivos en formato PDF sin ningún requerimiento adicional. Es gratuita, y su licencia permite que sea modificada libremente.

Entre las funcionalidades que nos ofrece esta librería nos encontramos:

- Elección de la unidad de medida, formato de página y márgenes
- Gestión de cabeceras y pies de página
- Salto de página automático
- Salto de línea y justificación del texto automáticos
- Admisión de imágenes (JPEG , PNG y GIF (versión 1.6))
- Colores
- Enlaces

Esta librería es desarrollada orientada a objetos, siendo el Objeto fpdf el encargado de ir almacenando la estructura, y mostrandolo con la funcion Output, teniendo diferentes salidas tanto por pantalla como por impresora o simplemente ofreciendo la posibilidad descargar el archivo. Fpdf ofrece la ventaja de permitir crear pdf desde php con un relativa sencillez haciendo de intermediario entre las funciones elementales de salida de datos que pintan el pdf y el usuario, entre sus funciones más utilizadas se encuentra Cell que es la base de todo el muestreo, creando celdas las cuales pueden contener texto.

HTML2PDF

Html2pdf Converter es un convertidor de HTML a PDF escrito en PHP4, utilizando la biblioteca FPDF de Olivier PLATHEY.

Permite la conversión de HTML 4.01 válido en formato PDF, y se distribuye bajo licencia **GPL.**

Esta biblioteca ha sido diseñada principalmente para manejar tablas para generar las facturas, entrega, y otros documentos oficiales. Aún no cuenta con todas las etiquetas, y el flotador.

PHPMailer

PHPMailer es una clase escrita en PHP que facilita el envío de correo, añadiendo facilidad en el envío de correos con adjuntos, en formato HTML y con diferentes codificaciones, soporte para imágenes embebidas, headers personalizados y además funciona con múltiples servidores de correo.

Típicamente para el envío de correo con PHP se utiliza la función ***mail()***, pero esta función tiene varias limitaciones, por ejemplo que no soporta el envío de adjuntos. Entonces PHPMailer nos viene a facilitar este trabajo que de otra forma sería muy engorroso. Viene con un conjunto de métodos que nos ayudarán en el envió de emails.

Ldap

Librería que permite la conexión al LDAP desde PHP y gestionar usuarios, etc.

CI-Template

La biblioteca CI-Template, escrita para el framework de PHP CodeIgniter, permite el manejo de vistas de una forma más sencilla y organizada. Fue creada en respuesta a la petición de la comunidad de CodeIgniter, de cómo optimizar el trabajo con múltiples vistas desde una controladora y cómo embeber vistas dentro de vistas de una manera estándar.

Es válido utilizar la biblioteca Template si:

- Se dificulta el manejo de múltiples vistas, especialmente a la hora de embeberlas.
- Se quiere evitar llamar las vistas header, footer y demás vistas globales desde cada controladora
- Es preferible tener una plantilla principal que puede cambiarse de manera general para todas las controladoras, de modo que pueda aplicarse a la aplicación un diseño uniforme.

CI-Message

Gestiona la mensajería del sistema. Brinda un conjunto de funciones para generar mensajes en diferentes formatos, con diferentes propósitos, sea la comunicación del éxito de una operación o la comisión de algún error por parte del usuario. Los mensajes están ya formateados con CSS (Cascading Style Sheets), adecuándose al fin para el que son generados.

Json

Gestiona la conversión y des conversión de cadenas Json para uso en el JavaScript. JSON es un formato de intercambio de datos creado a partir de un subconjunto de la notación de literales de objetos en JavaScript. Aunque JavaScript acepta una sintaxis para valores literales muy flexible, es importante tener en cuenta que JSON posee reglas mucho más estrictas.

Graphviz

Sitio oficial (http://www.graphviz.org/), software de visualización gráfica de código abierto que dispone de varios programas de diseño gráfico, de interfaces gráficas interactivas en línea, y herramientas auxiliares, además de librerías para diferentes lenguajes. Toma descripciones de gráficos en un lenguaje de texto

sencillo y a partir de este, representa diagramas en varios formatos útiles de imagen, como SVG para páginas web y PostScript para su inclusión en PDF u otros documentos, los cuales pueden ser mostrados en un navegador gráfico interactivo.

Tiene muchas funciones útiles para los esquemas concretos, como las opciones de colores, fuentes, cuadros, nodo, diseños, estilos de línea, los hipervínculos y formas personalizadas.

El comando de Graphviz comúnmente empleado es "dot" (punto), que representa dibujos jerárquicos o capas de dibujos dirigidos. El algoritmo de representación permite colocar bordes en la misma dirección (de arriba a abajo, o de izquierda a derecha) e intenta evitar los cruces y reducir la longitud de las aristas.

Luego de contar con una versión casi definitiva del diseño, haber decidido qué componentes utilizar y qué estrategia de integración tomar en consideración, durante la implementación del sistema, para dar solución a algunos problemas los algoritmos cobran cierto nivel de complejidad, por esta razón se examinan en el siguiente epígrafe esos algoritmos.

2.3 Descripción de los algoritmos no triviales a implementar. Análisis de complejidad de los mismos.

Los algoritmos que se analizan a continuación tienen un nivel de complejidad superior a la media, se hace necesario su estudio porque brindan solución a problemáticas claves en la implementación del sistema, como son reportes gráficos, búsqueda avanzada, entre otros.

Reporte gráfico de relación entre elementos

Consiste en la generación de un reporte que muestre un gráfico con las relaciones que existen entre dos tipos de elementos de una metodología.

Para su implementación fue necesaria la utilización de un algoritmo que partiendo de la entrada de los datos:

- *identificador de la metodología*
- *identificador del tipo de elemento* que será representado en el eje x
- *identificador del tipo de elemento* que será representado en el eje y

se determina una relación entre los elementos seleccionados, esta relación es un número entero entre 1 y 10, que son las combinaciones posibles entre todos los elementos, estas son:

1. Fase – Disciplina
2. Fase – Actividad
3. Fase – Rol
4. Fase – Artefacto
5. Disciplina – Actividad
6. Disciplina – Rol
7. Disciplina – Artefacto
8. Actividad – Rol
9. Actividad – Artefacto
10. Rol – Artefacto

De acuerdo a la relación determinada por la selección de ambos tipos de elementos y la metodología, se realiza una consulta a la Base de Datos, la cual retorna un arreglo con las relaciones existentes entre ellos, mostrándose un gráfico similar al que se muestra en la siguiente figura.

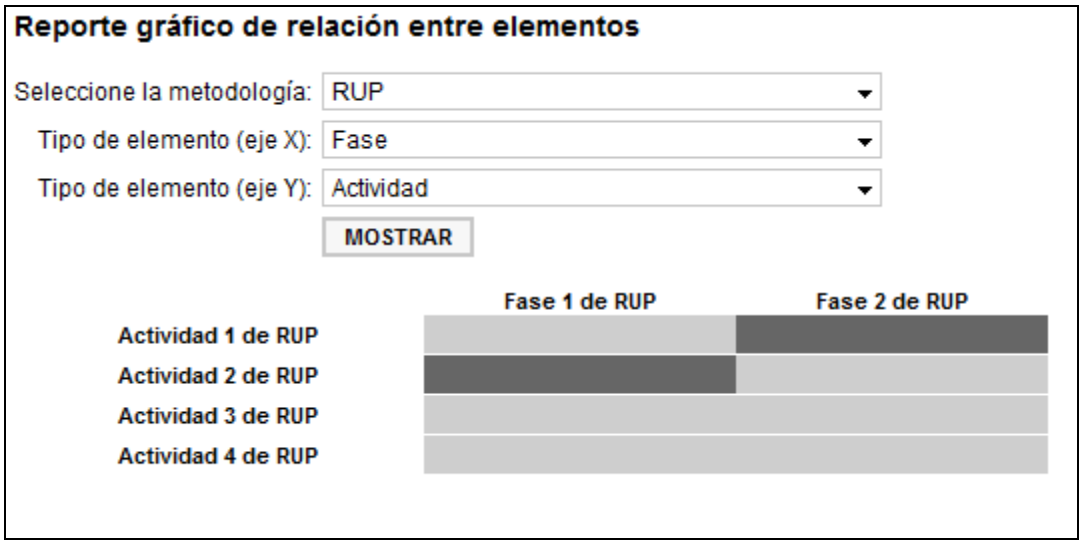


Figura 3. Ejemplo de reporte gráfico de relación entre elementos de una metodología de desarrollo.

Otro de los algoritmos no triviales a estudiar es uno cuya utilización en el sistema es prácticamente crítica, por el número de casos de uso que lo incluyen, la búsqueda avanzada.

Búsqueda avanzada

Consiste en la realización de una búsqueda de todos los elementos que en su nombre o descripción contengan la palabra especificada, siempre y cuando tenga al menos la cantidad de letras establecidas en el archivo de configuración, y en caso de que se especifique también un tipo de elemento, ya sea Fases, Disciplinas, Actividades, Roles o Artefactos, entonces la búsqueda se realizará solo en estos tipos de elementos teniendo en cuenta la metodología seleccionada.

Este cuenta además con un paginado dinámico permitiendo que los datos obtenidos se muestren según la cantidad especificada en el archivo de configuración, que contiene el número de elementos por página que deben mostrarse. Debe examinarse también el algoritmo que permite representar gráficamente un flujo de trabajo, este algoritmo tiene una elevada complejidad.

Flujo de trabajo

Consiste en la representación gráfica del flujo de trabajo de una fase contenida en una metodología de desarrollo de software.

Para la implementación de esta funcionalidad se hizo uso de Graphviz, explicada en el epígrafe anterior.

La mayoría de las herramientas case manejan los conceptos de Unión, Separación, Condicional, Inicio y Fin de forma gráfica, es decir cuentan con un editor gráfico para la creación de los diagramas. En el sistema no se hace de forma gráfica, sino mediante la descripción de las relaciones que existen entre las actividades.

La representación debe optimizar el espacio y hacer el diagrama comprensible al mismo tiempo, parte de esto lo garantizan los algoritmos que implementa Graphviz, pero no es capaz de determinar cuándo aplicar los conceptos de Inicio, Fin, Unión, Separación y Condicional.

Para eso se desarrolla un algoritmo que procesa todas las actividades y sus relaciones archivadas:

- Haciendo que las actividades que no tengan dependencia de otra actividad, lo hagan del Inicio.

- Representando el Final dependiendo de las actividades que no llevan a ninguna otra.
- Agrupando las actividades que llevan hacia un mismo destino en una Unión.
- Colocando una Separación en todas aquellas actividades que generan otras actividades.
- Uniendo en una Condicional aquellas actividades que llevan a otras mediante una relación verdadera o falsa.

¿Cómo ocurre la interacción?: En la clase Diagrama, la información del sistema se modela en forma de Actividades, Relaciones (relaciones simples) y Relaciones Condicionales, esta información es procesada para determinar el Inicio, Fin, Uniones, Separaciones y Condicionales del diagrama. Cada una de estas clases implementa una interfaz para generar el código "dot" del elemento que contiene. Con el código "dot" del diagrama se almacena en un archivo temporal y se ejecuta el comando de sistema "dot" pasándole como parámetro ese archivo de texto que contiene la descripción del grafo, el formato de imagen deseado y la dirección donde se desea guardar la salida.

Los algoritmos analizados anteriormente no pueden verse aislados del sistema, sino como funcionalidades de alguna clase, de las muchas con las que cuenta la aplicación. En el siguiente epígrafe se realizará una descripción de las nuevas clases implementadas para dar solución a la situación problemática.

2.4 Descripción de las nuevas clases u operaciones necesarias.

Siguiendo el enfoque MVC del CodeIgniter, las nuevas clases a describir caen en alguna de las categorías de Controladora, Modelo o Vista (Interfaz). A continuación las controladoras:

Nombre: Metodología	
Tipo de clase: Controladora	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	crear
Descripción:	Permite la creación de una metodología de desarrollo

Nombre:	listar
Descripción:	Muestra un listado con las metodologías existentes
Nombre:	buscar
Descripción:	Busca una metodología a través de un criterio dado
Nombre:	modificar_elementos
Descripción:	Modifica los elementos de una metodología
Nombre:	eliminar
Descripción:	Permite eliminar una metodología
Nombre:	activar
Descripción:	Permite activar una metodología de desarrollo
Nombre:	pdf
Descripción:	Genera un archivo PDF con datos de una metodología especificada
Nombre:	validar
Descripción:	Valida que los elementos de una metodología especificada sean correctos
Nombre:	habilitar
Descripción:	Habilita una metodología especificada
Nombre:	deshabilitar
Descripción:	Deshabilita una metodología especificada

Nombre:	flujo_trabajo_diagrama
Descripción:	Muestra el diagrama de flujo de trabajo de una metodología especificada
Nombre:	asociar_elementos
Descripcion:	Permite asociar elementos de una metodología a otra
Nombre:	asociar_proyecto
Descripcion:	Permite asociar un proyecto a una metodología
Nombre:	terminar edición
Descripcion:	Termina la edición de una metodología especificada

Nombre: Fase	
Tipo de clase: Controladora	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	crear
Descripción:	Permite la creación de una nueva fase en una determinada metodología
Nombre:	listar
Descripción:	Muestra un listado con las fases existentes en una metodología.
Nombre:	buscar
Descripción:	Busca una fase a través de un criterio dado

Nombre:	modificar
Descripción:	Modifica los datos de una fase especificada
Nombre:	eliminar
Descripción:	Permite eliminar una fase
Nombre:	detalles
Descripción:	Muestra los detalles de una fase especificada

Esta misma descripción es similar para el caso de otras clases controladoras las cuales presentan las mismas funcionalidades, entre estas se encuentran las clases: Disciplina, Actividad, Rol y Artefacto.

Nombre: Foro	
Tipo de clase: Controladora	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	agregar
Descripción:	Permite agregar un nuevo foro
Nombre:	modificar
Descripción:	Permite modificar los datos de un foro
Nombre:	eliminar
Descripción:	Permite eliminar un foro especificado
Nombre:	mostrar
Descripción:	Muestra los temas de un foro especificado

Nombre: Usuario	
Tipo de clase: Controladora	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	ws
Descripción:	Muestra el listado de todos los proyectos a los que está asociado el usuario
Nombre:	entrar
Descripción:	Valida que el usuario y la clave para entrar al sistema sean correctos
Nombre:	registrar
Descripción:	Registra un nuevo usuario en el sistema
Nombre:	salir
Descripción:	Destruye la sesión del usuario que estaba logueado
Nombre:	registrar_empresa
Descripción:	Permite a un usuario jurídico insertar una empresa en el sistema
Nombre:	mostrar_empresa
Descripción:	Muestra los datos de la empresa de un usuario determinado
Nombre:	modificar_empresa
Descripción:	Modifica los datos de la empresa a la que pertenece un usuario determinado

Nombre:	perfil
Descripción:	Muestra los datos de perfil de un usuario determinado
Nombre:	modificar
Descripción:	Modifica los datos de un usuario determinado
Nombre:	cambiar_clave
Descripción:	Cambia la clave de un usuario determinado
Nombre:	recuperar_clave
Descripción:	Recuperara la clave de un usuario determinado

Nombre: Reporte	
Tipo de clase: Controladora	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	registro_exitoso
Descripción:	Muestra un reporte de los usuarios registrados exitosamente
Nombre:	rol_proyecto_usuario
Descripción:	Muestra un reporte de los usuarios vinculados a proyectos
Nombre:	proyecto_metodologia
Descripción:	Muestra un reporte de los proyectos por metodología
Nombre:	acciones_elemento
Descripción:	Muestra un reporte de las acciones realizadas sobre un elemento

Nombre:	artefacto_rol
Descripción:	Muestra un reporte de los artefactos generados por un rol determinado
Nombre:	grafico
Descripción:	Muestra un grafico de la relación que existe entre dos elementos
Nombre:	actividad_disciplina
Descripción:	Muestra un reporte de las actividades realizadas sobre una disciplina de una metodología determinada
Nombre:	actividad_fase
Descripción:	Muestra un reporte de las actividades realizadas sobre una fase de una metodología determinada
Nombre:	artefacto_actividad
Descripción:	Muestra un reporte de los artefactos generados que ejecutan una actividad de una metodología determinada
Nombre:	actividades
Descripción:	Muestra un reporte de los actividades que contienen subactividades de una metodología determinada

Nombre: Encuesta	
Tipo de clase: Controladora	
Atributos:	**Tipo:**
-	-

Funciones:	
Nombre:	listar
Descripción:	Muestra el listado de las encuestas
Nombre:	resultado
Descripción:	Muestra los resultados de la encuesta
Nombre:	votar
Descripción:	Realiza una votación sobre una encuesta determinada

Estas son las clases modelos:

Nombre: Metodología	
Tipo de clase: Modelo	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	obtener_proceso (id_proceso)
Descripción:	Retorna los datos del proceso especificado
Nombre:	obtener_proyectos (id_metodologia)
Descripción:	Retorna los proyectos que pertenecen a la metodología especificada
Nombre:	obtener_fases_metodologia (id_metodologia)
Descripción:	Retorna las fases que pertenecen a la metodología especificada
Nombre:	obtener_disciplinas_metodologia (id_metodologia)
Descripción:	Retorna las disciplinas que pertenecen a la metodología

	especificada
Nombre:	obtener_actividades_metodologia (id_metodologia)
Descripción:	Retorna las actividades que pertenecen a la metodología especificada
Nombre:	obtener_artefactos_metodologia (id_metodologia)
Descripción:	Retorna los artefactos que pertenecen a la metodología especificada
Nombre:	artefacto_asociado_rol (id_artefacto)
Descripción:	Verifica si el artefacto cuyo identificador recibe por parámetro está asociado con al menos 1 rol
Nombre:	artefacto_asociado_actividad (id_artefacto)
Descripción:	Verifica si el artefacto cuyo identificador recibe por parámetro está asociado con al menos 1 actividad
Nombre:	obtener_roles_metodologia (id_metodologia)
Descripción:	Retorna los roles que pertenecen a la metodología especificada
Nombre:	rol_asociado_actividad (id_rol)
Descripción:	Verifica si el rol cuyo identificador recibe por parámetro está asociado con al menos 1 actividad
Nombre:	actividad_asociada_fase (id_actividad)
Descripción:	Verifica si la actividad cuyo identificador recibe por parámetro está asociada con al menos 1 fase

Nombre:	actividad_asociada_disciplina (id_actividad)
Descripcion:	Verifica si la actividad cuyo identificador recibe por parámetro está asociada con al menos 1 disciplina
Nombre:	disciplina_asociada_fase (id_disciplina)
Descripcion:	Verifica si la disciplina cuyo identificador recibe por parámetro está asociada con al menos 1 fase
Nombre:	asociar_elemento_metodologia (tabla, id, id_metodologia)
Descripcion:	Asocia un element ya existenete a la metodología especificada
Nombre:	es_metodologia_editable (id_metodologia)
Descripcion:	Retorna si la metodologia es editable
Nombre:	usuario_permiso_metodologia(id_usuario, id_met, habilitadas)
Descripcion:	Retorna si un usuario tiene permisos sobre una metodologia
Nombre:	terminar_edicion (id_metodologia)
Descripcion:	Termina la edición de la metodologia

Nombre: Fase	
Tipo de clase: Modelo	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	Crear (datos)
Descripción:	Adiciona una fase a una metodología especificada

Nombre:	listar
Descripción:	Muestra un listado con las fases existentes
Nombre:	Buscar (criterio)
Descripción:	Busca una fase a través de un criterio dado
Nombre:	Modificar (id_fase)
Descripción:	Modifica los datos de una fase especificada
Nombre:	Eliminar (id_fase)
Descripción:	Elimina una fase especificada
Nombre:	Detalles (id_fase)
Descripción:	Muestra los detalles de una fase especificada

Esta descripción es similar para el caso de otras clases modelos las cuales presentan las mismas funcionalidades, entre estas se encuentran las clases: Disciplina, Actividad, Rol y Artefacto.

Nombre: Foro	
Tipo de clase: Modelo	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	Agregar (datos)
Descripción:	Agrega un nuevo foro
Nombre:	Modificar (id_foro)
Descripción:	Modifica los datos de un foro especificado

Nombre:	Eliminar (id_foro)
Descripción:	Elimina un foro especificado
Nombre:	Mostrar (id_foro)
Descripción:	Muestra los temas de un foro especificado

Nombre: Usuario	
Tipo de clase: Modelo	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	autenticar (usuario, clave)
Descripción:	Retorna verdadero si el usuario y la clave especificada son válidos para entrar al sistema.
Nombre:	existe_usuario (usuario)
Descripción:	Retorna verdadero si el usuario especificado existe.
Nombre:	existe_correo (correo)
Descripción:	Retorna verdadero si ya existe un usuario con la dirección de correo especificada.
Nombre:	registrar_juridico (id_usuario, id_cargo)
Descripción:	Retorna verdadero si se pudo registrar el usuario especificado.
Nombre:	obtener_usuario (usuario)
Descripción:	Retorna todos los datos del usuario especificado.

Nombre:	obtener_cargo(usuario)
Descripción:	Retorna el cargo del usuario especififcado.
Nombre:	con_permiso_metodologia (id_metodologia)
Descripción:	Retorna todos los usuarios con permisos sobre la metodología especificada.
Nombre:	deshabilitar (id_usuario)
Descripción:	Deshabilita el usuario especificado, prohibiendo su acceso al sistema y retorna verdadero si se pudo realizar la operación.
Nombre:	habilitar (id_usuario)
Descripción:	Habilita el usuario especificado, otorgando a este acceso al sistema y retorna verdadero si se pudo realizar la operación.
Nombre:	cambiar_clave (usuario, nueva_clave)
Descripción:	Retorna verdadero si se cambio con éxito la clave del usuario especificado.
Nombre:	modificar_usuario (datos)
Descripción:	Retorna verdadero si se actualizaron correctamente los datos del usuario especificado.
Nombre:	eliminar_usuario (id_usuario)
Descripción:	Retorna verdadero si se eliminó con éxito el usuario especificado.

Nombre: Reporte
Tipo de clase: Modelo

Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	registro_exitoso ()
Descripción:	Retorna los usuarios registrados exitosamente en rango de fechas
Nombre:	rol_proyecto_usuario (id_usuario, id_proyecto)
Descripción:	Retorna los roles de un usuario en un determinado proyecto.
Nombre:	proyecto_metodologia (id_metodologia)
Descripción:	Retorna los proyectos de la metodología especificada
Nombre:	acciones_elemento (id_elemento)
Descripción:	Rertorna las acciones realizadas sobre un elemento determinado.
Nombre:	artefacto_rol
Descripción:	Muestra un reporte de los artefactos generados por un rol determinado
Nombre:	grafico (id_elemento1, id_elemento2)
Descripción:	Retorna las relaciones que existen entre dos tipos de elementos
Nombre:	actividad_disciplina (id_metodologia, id_disciplina)
Descripción:	Retorna todas las actividades realizadas sobre una disciplina de una metodología determinada
Nombre:	actividad_fase (id_metodologia, id_fase)
Descripción:	Retorna todas las actividades realizadas sobre una fase de una metodología determinada

Nombre:	artefacto_actividad (id_metodologia, id_actividad)
Descripción:	Retorna los artefactos generados que ejecutan una actividad de una metodología determinada
Nombre:	actividades (id_metodologia)
Descripción:	Retorna las actividades que contienen subactividades de una metodología determinada

Nombre: Encuesta	
Tipo de clase: Modelo	
Atributos:	**Tipo:**
-	-
Funciones:	
Nombre:	listar
Descripción:	Retorna un listado de las encuestas existentes
Nombre:	resultado
Descripción:	Retorna los resultados de la encuesta
Nombre:	Votar (id_encuesta)
Descripción:	Retorna verdadero si se pudo realizar la votación sobre una encuesta determinada

Estas son las clases interfaz:

Nombre: Metodología
Tipo de clase: Interfaz
Descripción General: Muestra las interfaces necesarias para crear una nueva metodología, modificar una existente, mostrar un listado de estas, realizar búsquedas sobre estas, asociar elementos a una metodología, asociar a un proyecto existente, consultar sus elementos, exportarlas a PDF, habilitar o deshabilitar una metodología determinada, así como una interfaz en la que se muestra un diagrama con el flujo de trabajo.

Nombre: Fase
Tipo de clase: Interfaz
Descripción General: Muestra las interfaces necesarias para crear una nueva fase, modificar o eliminar una existente, mostrar un listado de estas, y realizar búsquedas sobre estas.

Esta descripción es similar para el caso de otras clases interfaces las cuales presentan las mismas funcionalidades, entre estas se encuentran las clases: Disciplina, Actividad, Rol y Artefacto.

Nombre: Reporte
Tipo de clase: Interfaz
Descripción General: Muestra un conjunto de interfaces las cuales muestran los resultados de varios reportes que brinda el sistema, entre estos se pueden mencionar: Reporte de los usuarios registrados exitosamente, Reporte de los

usuarios vinculados a proyectos, Reporte de los proyectos por metodología, Reporte de las acciones realizadas sobre un elemento, Reporte gráfico de la relación que existe entre dos elementos, Reporte de los actividades que contienen subactividades de una metodología determinada, entre otros.

Nombre: Usuario
Tipo de clase: Interfaz
Descripción General: Muestra las interfaces necesarias para inscribir un nuevo usuario al sistema, modificar o eliminar uno existente, mostrar un listado de estos, habilitarlos o deshabilitarlos, recuperar su clave en caso de que la haya olvidado, modificar los datos personales.

Nombre: Foro
Tipo de clase: Interfaz
Descripción General: Muestra las interfaces para inscribir un nuevo foro, modificar o eliminar uno existente, agregar un tema, así como responder una respuesta sobre un tema determinado.

Nombre: Encuesta
Tipo de clase: Interfaz
Descripción General: Muestra una interfaz con una encuesta, y posteriormente a votar muestra los resultados de dicha encuesta.

Después de realizar una valoración crítica del diseño propuesto por el analista, se ha podido arribar a un diseño acorde con las restricciones del sistema, que deben tenerse en consideración a la hora de implementarlo, a dicha propuesta se le realizó un conjunto de cambios adaptativos que permiten implementar el sistema de forma más eficiente, aunque los cambios no fueron globales porque la estructura básica, dígase las clases y demás elementos que componen un diagrama de clases del diseño, se aceptaron, lo que varió fue la manera de relacionarlos.
También se realizó un análisis a un conjunto de librerías y clases previamente implementadas a reutilizar, que repercuten en la dinámica de implementación porque brindan facilidades que es solamente cuestión de utilizar, entre ellas la estructura del framework de PHP a emplear, CodeIgniter.

De igual manera se estudiaron los algoritmos no triviales presentes en la solución, realizándole a cada uno un análisis de su complejidad y se describieron las nuevas clases y funciones necesarias para solucionar el problema, estas clases según el patrón de diseño que implementa CodeIgniter, se clasificaron en tres grupos básicos: controladoras, modelos y vistas (interfaces). Todas estas cuestiones tienen una repercusión directa en la consumación de la aplicación.

CAPÍTULO 3. Validación de la solución propuesta.

La prueba del software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

De ahí que el presente capítulo tenga como objetivo, realizar una búsqueda o diseño de las pruebas unitarias que permitirán validar la solución propuesta, realizando además una descripción de estas, tomando en consideración el objetivo de la prueba, su alcance, tipo y detalles. Posteriormente se describirán los valores empleados para las pruebas y para concluir una evaluación de su ejecución y los resultados obtenidos.

3.1 Búsqueda o diseño de los test de unidades que permitan validar la solución propuesta.

Según la ayuda especializada del Rational Rose la prueba de unidad es la prueba enfocada a los elementos probables más pequeños del software, consiste en una prueba estructural (o caja blanca), lo cual requiere conocer el diseño interno de la unidad puesto que verifica la lógica interna y el flujo de datos; y una prueba de especificación (de caja negra), basada sólo en la especificación del comportamiento externamente visible de la unidad. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. En la siguiente ilustración se muestran las pruebas que se dan como parte de la prueba de unidad, según Roger S. Pressman en Ingeniería del Software. Un enfoque práctico. Parte 1.

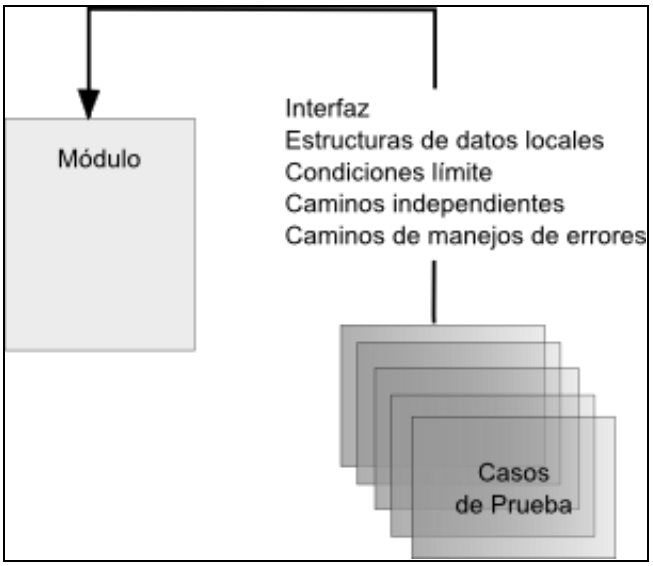


Figura 4. Prueba de Unidad

Primero se prueba la interfaz del módulo para garantizar un adecuado flujo de información hacia y desde la unidad que está siendo probada. Se comprueban las estructuras de datos locales para asegurar la integridad de los datos que se almacenan temporalmente durante toda la ejecución del algoritmo. También se prueba el módulo en condiciones límites para garantizar que funciona correctamente en los límites establecidos como restricciones de procesamiento. Se ejercitan todos los caminos independientes (caminos básicos), garantizando que todas las posibles vertientes se ejecuten por lo menos una vez, cuando debido a la complejidad estructural del objeto de la prueba esto no es posible, se seleccionan los caminos independientes con mayor probabilidad de ser ejecutados, y es a estos que se le practican las pruebas. Finalmente se prueban todos los caminos de manejos de errores.

Este trabajo desea añadir a la definición anterior de prueba unitaria, que en el contexto orientado a objetos (OO), los elementos probables más pequeños, en los que se enfocarían las pruebas de unidad serían las clases u objetos encapsulados. Una clase puede contener un cierto número de operaciones, y una operación particular puede existir como parte de un número de clases diferentes. Esta prueba de clases para el software OO es equivalente a la prueba de unidad para el software convencional.

La prueba de clases para software OO está dirigida por las operaciones encapsuladas por la clase y el estado del comportamiento de la clase. No se puede probar una operación aisladamente sino como parte

de una clase. Válido aclarar esto puesto que el sistema que se está implementando es completamente orientado a objetos.

Cuando se definía una prueba unitaria se mencionaron dos enfoques necesarios para la correcta conducción de las pruebas: caja blanca y caja negra:

- Prueba de Especificación o de Caja Negra: Tiene como propósito verificar las relaciones de entrada y salida de una unidad. Su objetivo es verificar qué hace la unidad, pero sin averiguar cómo lo hace. Se centra y basa en la entrada/salida de datos de las operaciones de la unidad.
- Prueba Estructural o de Caja Blanca: Verifica que la estructura interna de la unidad sea correcta. Mediante esta prueba el ingeniero del software puede obtener casos de prueba que:
    - Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
    - Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
    - Ejecuten todos los bucles en sus límites operacionales.
    - Ejerciten las estructuras internas de datos para asegurar su validez.

El diseño de casos de prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente.

No es objetivo de este capítulo profundizar en las pruebas de interfaz, pero es válido destacar que fue necesaria la realización de estas al software, para probar de esta forma la operatividad y validez de las funcionalidades del sistema, así como el rendimiento del mismo bajo ciertas condiciones de estrés, en las que se pudiera ver afectada la aplicación, todas estas pruebas se realizaron tomando como base los requisitos funcionales del mismo.

Este capítulo se centra, a partir de todo el proceso anteriormente explicado, en la ejecución de las pruebas de caminos independientes, que es una prueba enfocada a Caja Blanca, una técnica inicialmente propuesta por Tom McCabe en 1976, que permitirá obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos

de ejecución. Se basa en construir un caso de prueba por camino básico que se encuentre en el grafo de programa asociado al método de la clase que se desea someter a pruebas. Los pasos del método son:

1. Se dibuja un grafo de flujo (o grafo del programa) G, que es una representación del flujo de control que si bien no es imprescindible para la conducción de la prueba, sirve como herramienta útil para ilustrar el método.

2. Se determina la complejidad ciclomática del grafo, V(G). Esta es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. En el contexto del método de prueba del camino básico, el valor calculado define el número de caminos independientes de un programa y proporciona un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. Se puede calcular de tres formas:

   - V(G) coincide con el número de regiones del grafo de flujo.
   - V(G) = A – N + 2 donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
   - V(G) = P + 1 donde P es el número de nodos predicado contenidos en el grafo de flujo. Un nodo predicado es cada nodo que contiene una condición y se caracteriza gráficamente porque dos o más aristas emergen de él.

3. Se construye una secuencia de V(G) caminos linealmente independientes en G. Recursivamente:

   a) El primer camino es cualquiera de los caminos mínimos del nodo origen del grafo a uno de los nodos de terminación del grafo. Se inicializa E (G), el conjunto de aristas en la secuencia linealmente independiente de G con todas aquellas aristas que forman este primer camino.

   b) El próximo camino se forma agregando un nuevo camino entre el origen y una terminación de G Este nuevo camino debe agregar el mínimo número posible de aristas nuevas a E, (pero que agregue por lo menos una arista nueva).

4. Se prepara un caso de prueba por camino hallado en el paso anterior.

   a) Determinar los datos a proporcionar como entrada para ejecutar el camino hallado.

b) Usando la especificación funcional del método, indicar cuál es el resultado esperado.

La siguiente imagen ilustra la notación del grafo de flujo.

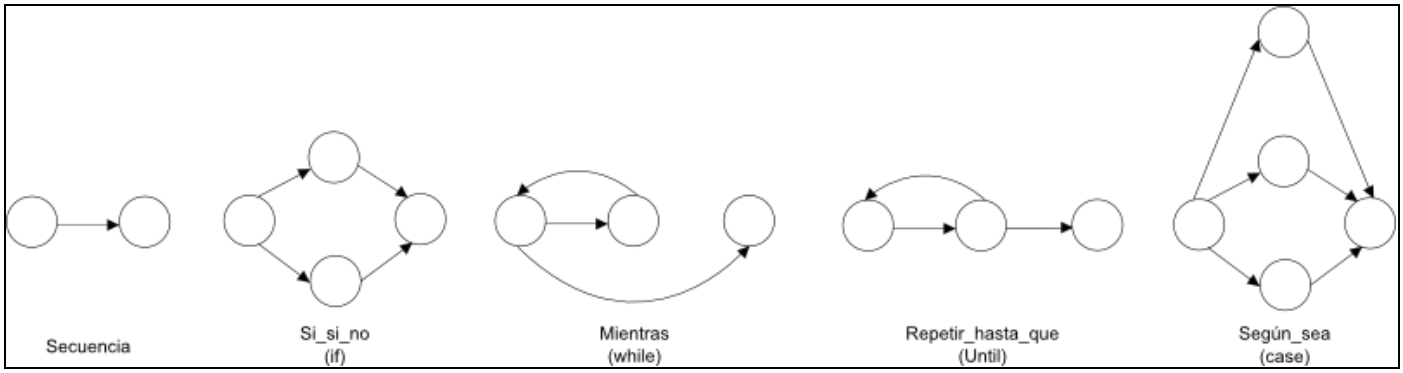


Figura 5. Notación de grafo de flujo

Este último paso conlleva a otro de los aspectos relevantes a tratar en el presente capítulo, la descripción de los valores utilizados para las prueba.

Los datos se escogieron de forma que las condiciones de los nodos predicados, estén adecuadamente establecidas, con el propósito de comprobar cada camino.

Después de haber determinado el método de prueba, se procede a la ejecución de esta, para lo cual se diseñó una clase que tuviese las funcionalidades necesarias para probar cada uno de los casos de prueba a tratar.

A continuación se muestra la clase encargada de conducir las pruebas, luego las funciones que van a someterse a prueba, a las que se le construyó su grafo de flujo de datos, se le calculó la complejidad ciclomática, se determinó el conjunto de caminos independientes, posteriormente se muestra el juego de datos seleccionado, y para concluir una tabla resumen con las entradas de datos al sistema por parte del usuario, así como el resultado esperado y las respuestas del sistema correspondientes a estas.

Dada la complejidad y extensión de las pruebas requeridas para validar la solución propuesta en el presente capítulo se documentan, como patrón, las que se aplicaron a las principales funcionalidades del

módulo metodología, específicamente las funciones de la clase librería, que son las que median entre el flujo de datos (clases controladoras) y la persistencia (clases modelos), por lo que tienen mayor relevancia en el sistema. Los demás módulos fueron sometidos a pruebas similares.

Los casos de prueba

Clase de prueba

```php
<?php
class Prueba extends Controller {

	public function __construct()
	{
		parent::Controller();

		// Cargar librería Metodologia
		$this->load->module_library('metodologia', 'metodologia_lib');

		$this->datos_metodologia = array(
			'nombre'		=> 'Rational Unified Process',
			'identificador' => 'RUP',
			'principios'	=> 'Iterativo e Incremental',
			'proceso'		=> '2',
			'usuarios'		=> array('1','2')
		);
	}

	public function verificar_agregar()
	{
		$this->metodologia_lib->agregar($this->datos_metodologia);

		$identificador = $this->datos_metodologia['identificador'];

		$datos = $this->metodologia_lib->obtener_por_id($identificador);

		return count(array_diff_assoc($datos, $this->datos_metodologia)) == 0;
	}

	public function verificar_modificar()
	{
		$this->metodologia_lib->agregar($this->datos_metodologia);

		$identificador = $this->datos_metodologia['identificador'];

		$datos = $this->metodologia_lib->obtener_por_id($identificador);

		if (count(array_diff_assoc($datos, $this->datos_metodologia)) == 0)
		{
			$this->datos_metodologia['nombre'] = "Scrum";

			$this->metodologia_lib->agregar($this->datos_metodologia);

			$datos = $this->metodologia_lib->obtener_por_id($identificador);
```

```
                return $datos['nombre'] == "Scrum";
        }

        return FALSE;
    }

    public function verificar_eliminar()
    {
        $this->metodologia_lib->agregar($this->datos_metodologia);

        $identificador = $this->datos_metodologia['identificador'];

        $this->metodologia lib->eliminar($identificador);

        $datos = $this->metodologia_lib->obtener_por_id($identificador);

        return count($datos) == 0;
    }

    public function verificar_habilitar($id_metodologia)
    {
        $identificador = $this->datos_metodologia['identificador'];

        $this->metodologia_lib->habilitar($identificador);

        $datos = $this->metodologia_lib->obtener_por_id($identificador);

        return $datos['activa'] == TRUE;
    }
}
?>
```

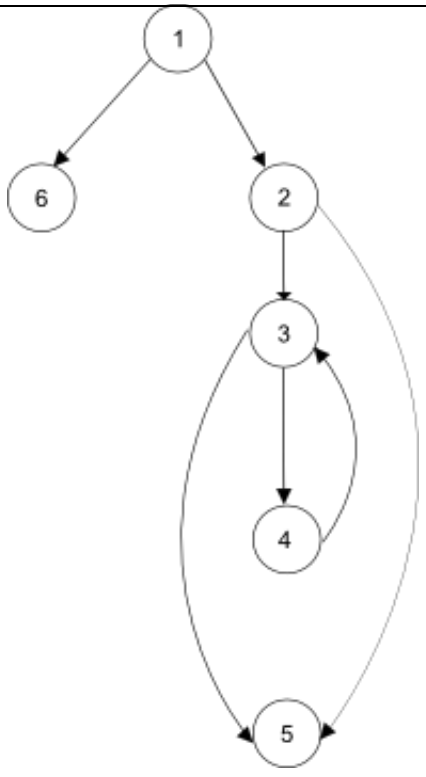
Funciones a probar

Descripción del CU Crear Metodología

Permite crear una metodología de desarrollo de software.

```
public function agregar($datos)
{
    $datos += array(
        'fecha_creada' => date("Y-m-d H:i:s"),
        'id_usuario'   => get_id_usuario()
    );

    $id_metodologia = $this->_ci->metodologia_mdl->create($datos);

    if ((int)$id_metodologia > 0)
    {
        if (isset($datos['usuario_metodologia']) &&
is_array_fill($datos['usuario_metodologia']))
        {
```

```
                    foreach($datos['usuario_metodologia'] as $id_usuario)
                    {
                        $this->_ci->metodologia_mdl->agregar_usuario_metodologia($id_usuario,
$id_metodologia);
                    }
            }

            $this->_ci->metodologia_mdl->agregar_usuario_metodologia(get_id_usuario(),
$id_metodologia);

            set_message('Se ha agregado la metodología', 'info');

            $this->_ci->session->set('hm_id_metodologia', $id_metodologia);
            $this-> ci->session->set('hm nombre', $datos['nombre']);

            return TRUE;
        }

        set message('No se ha agregado la metodología', 'error');
        return FALSE;
}
```

Complejidad ciclomática V (G) = 3	
Posibles caminos 1 – 6 1 – 2 – 3 – 5 1 – 2 – 3 – 4 – 3…	

Definir juego de datos

```
// Camino 1 - 2 - 3 - 4 - 3 - 5 OK
$juego_datos1 = array(
       'nombre'            => 'Rational Unified Process',
       'identificador' => 'RUP',
       'principios'  => 'Iterativo e Incremental',
       'proceso'           => '2',
       'usuarios'          => array('1','2')
);

// Camino 1 - 6 !OK
$juego_datos2 = array(
       'nombre'            => '',              // No puede ser una cadena vacía
       'identificador' => 'RUP',
       'principios'  => 'Iterativo e Incremental',
       'proceso'           => 'xxx',    // Debe ser un número entero
       'usuarios'          => array()   // No puede ser un arreglo vacío
);

// Camino 1 - 2 - 3… OK
$juego_datos3 = array(
       'nombre'            => 'Rational Unified Process',
       'identificador' => 'RUP',
       'principios'  => 'Iterativo e Incremental',
       'proceso'           => '2',
       'usuarios'          => array(-1)
);
```

CPR 1: Crear Metodología (camino 1)

Descripción de la Funcionalidad:

El usuario entra los datos de la metodología de software a crear, en caso de existir una metodología con el identificador especificado el sistema muestra un error, de lo contrario queda creada la metodología.

Flujo Central:

- El usuario da click en la opción del menú “Metodología/Crear”.
- El usuario entra los datos de la metodología que desea crear.
- El usuario da click en el botón “AGREGAR”.
- Se crea la metodología.

Condiciones de Ejecución:

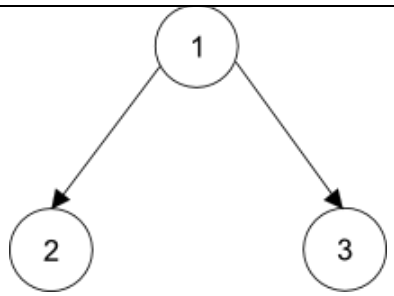
Los datos de la metodología deben ser válidos y no debe existir una metodología con el identificador introducido.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Obs.
El usuario entra datos válidos para crear una metodología.		El sistema muestra un mensaje de éxito de la operación.	El sistema muestra el mensaje: "Se ha agregado la metodología".	
	El usuario no especifica el nombre de la metodología a crear.	El sistema muestra un mensaje de error.	El sistema muestra el mensaje: "Debe entrar un nombre válido".	

Descripción del CU Eliminar Metodología

Permite eliminar una metodología de desarrollo de software.

```
public function eliminar($id_metodologia)
{
	if ($this->_ci->metodologia_mdl->delete($id_metodologia))
	{
		set_message('Se ha eliminado la metodología', 'info');
	}
	else
	{
		set_message('No se ha podido eliminar la metodología', 'error');
	}
}
```

Complejidad ciclomática V (G) = 2	
Posibles caminos 1 – 2 1 – 3	

Definir juego de datos

```
// Camino 1 - 2 OK
$id_metodologia = 1;  // Metodología válida

// Camino 1 - 3 !OK
$id_metodologia = -1;   // Metodología inválida
```

CPR 1: Eliminar Metodología (camino 1)

Descripción de la Funcionalidad:

El usuario selecciona la metodología a eliminar especificando el identificador de la misma, en caso de no existir el sistema muestra un error, de lo contrario queda eliminada la metodología.

Flujo Central:

- El usuario da click en la opción del menú “Metodología/Listar”.
- El usuario selecciona la metodología que desea eliminar.
- El usuario confirma que desea eliminar la metodología.
- Se elimina la metodología.

Condiciones de Ejecución:

El identificador de la metodología debe existir.

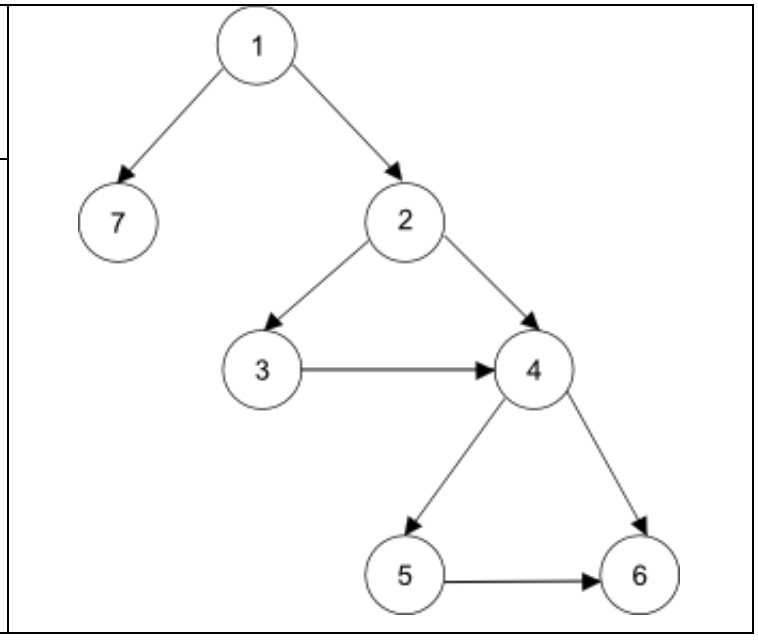
Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Obs.
El usuario entra un identificador de metodología válido.		El sistema muestra un mensaje de éxito de la operación.	El sistema muestra el mensaje: “Se ha eliminado la metodología”.	
	El usuario entra un identificador de metodología inválido.	El sistema muestra un mensaje de error.	El sistema muestra el mensaje: “Metodología no válida”.	

Descripción del CU Modificar Metodología

Permite modificar los datos de una metodología de desarrollo de software.

```
public function modificar($id_metodologia, $datos)
{
	$datos += array(
		'id_usuario'	=> get_id_usuario()
	);
	if ($this->_ci->metodologia_mdl->update($id_metodologia, $datos))
	{
		if (!(isset($datos['usuario_metodologia']) &&
is_array_fill($datos['usuario_metodologia'])))
		{
```

```
                        $datos['usuario_metodologia'] = array();
                }

                if (!in_array(get_id_usuario(), $datos['usuario_metodologia']))
                {
                        $datos['usuario_metodologia'][] = get_id_usuario();
                }

                $this->_ci->metodologia_mdl->modificar_usuario_metodologia($id_metodologia,
$datos['usuario_metodologia']);

                set_message('Se ha actualizado la metodología', 'info');
                return TRUE;
        }

        set_message('No se ha podido actualizar la metodología', 'error');

        return FALSE;
}
```

Complejidad ciclomática V (G) = 4	
Posibles caminos 1 – 7 1 – 2 – 4 – 6 1 – 2 – 3 – 4 – 6 1 – 2 – 4 – 5 – 6	

Definir juego de datos

```
// Camino 1 - 7 !OK
$id_metodologia = -1;  // Metodología inválida
$juego_datos1 = array(
        'nombre'            => 'Rational Unified Process (modificado)',
        'identificador' => 'RUP (modificado)',
        'principios'  => 'Iterativo e Incremental (modificado)',
        'proceso'           => '6',
        'usuarios'          => array('3','5')
```

```
);

// Camino 1 - 7 !OK
$id_metodologia = 1;  // Metodología válida
$juego_datos2 = array(
       'nombre'            => '',       // No puede ser una cadena vacía
       'identificador' => 'RUP',
       'principios'  => 'Iterativo e Incremental',
       'proceso'           => 'xxx',   // Debe ser un número entero
       'usuarios'          => array()  // No puede ser un arreglo vacío
);


// Camino 1 - 2 - 4 - 6 OK
$id_metodologia = 1;  //metodología válida

// Datos válidos
$juego_datos3 = array(
       'nombre'            => 'Rational Unified Process (modificado)',
       'identificador' => 'RUP (modificado)',
       'principios'  => 'Iterativo e Incremental (modificado)',
       'proceso'           => '6',
       'usuarios'          => array('3','5')
);
$datos['usuarios_metodologia'] = array(0 => 1); // Id del usuario logueado

// Camino 1 - 2 - 3 - 4 - 6 OK
$id_metodologia = 1;  // Metodología válida

//datos válidos
$juego_datos4 = array(
       'nombre'            => 'Rational Unified Process (modificado)',
       'identificador' => 'RUP (modificado)',
       'principios'  => 'Iterativo e Incremental (modificado)',
       'proceso'           => '6',
       'usuarios'          => array('3','5')
);
$datos['usuarios_metodologia'] = array();

// Camino 1 - 2 - 3 - 4 - 5 - 6 OK
$id_metodologia = 1;  // Metodología válida

//datos válidos
$juego_datos5 = array(
       'nombre'            => 'Rational Unified Process (modificado)',
       'identificador' => 'RUP (modificado)',
       'principios'  => 'Iterativo e Incremental (modificado)',
       'proceso'           => '6',
       'usuarios'          => array('3','5')
);
$datos['usuarios_metodologia'] = array();

// Camino 1 - 2 - 3 - 4 - 6 OK
$id_metodologia = 1;  // Metodología válida
$juego_datos5 = array(
       'nombre'            => 'Rational Unified Process (modificado)',
```

```
        'identificador' => 'RUP (modificado)',
        'principios'  => 'Iterativo e Incremental (modificado)',
        'proceso'           => '6',
        'usuarios'          => array('3','5')
);
$datos['usuarios_metodologia'] = array();
```

CPR 1: Modificar Metodología (camino 1)

Descripción de la Funcionalidad:

El usuario selecciona la metodología a modificar especificando el identificador de la misma, en caso de no existir el sistema muestra un error, de lo contrario el sistema muestra una vista con los datos de la metodología, el usuario modifica los datos de la metodología, en caso de que especifique un identificador ya existente el sistema muestra un error, de lo contrario queda modificada la metodología.

Flujo Central:

- El usuario da click en la opción del menú “Metodología/Listar”.
- El usuario selecciona la metodología que desea modificar.
- El usuario entra los datos de la metodología.
- El usuario da click en el botón “Modificar”.
- Queda actualizada la metodología.

Condiciones de Ejecución:

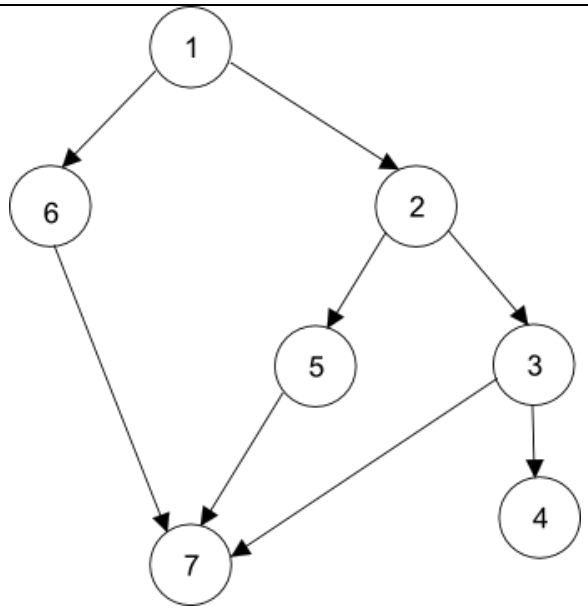
El identificador de la metodología debe existir y los datos de la metodología deben ser válidos.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Obs.
El usuario entra datos válidos para modificar una metodología.		El sistema muestra un mensaje de éxito de la operación.	El sistema muestra el mensaje: “Se ha actualizado la metodología”.	
	El usuario entra el identificador de una metodología que no existe.	El sistema muestra un mensaje de error.	El sistema muestra el mensaje: “ Metodología no válida”.	

Descripción del CU Habilitar Metodología

Permite habilitar una metodología de desarrollo de software.

```
public function habilitar($id_metodologia)
{
	$id_metodologia = (int)$id_metodologia;

	if ($id_metodologia > 0 && $this->_ci->metodologia_mdl->exist($id_metodologia))
	{
		if ($this->usuario_permiso_metodologia(get_id_usuario(), $id_metodologia, FALSE) ||
es_administrador(FALSE))
			{
			$datos = array('activa' => 1);

			if ($this->_ci->metodologia_mdl->update($id_metodologia, $datos))
			{
				set_message('Se ha habilitado la metodología');
				return TRUE;
			}
		}
		else
		{
			set_message('Usted no tiene permiso sobre esta metodología', 'error');
		}
	}
	else
	{
		set_message('Error al habilitar esta metodología', 'error');
	}

	return FALSE;
}
```

Complejidad ciclomática V (G) = 3	
Posibles caminos 1 – 6 – 7 1 – 2 – 3 – 4 1 – 2 – 5 – 7	

Definir juego de datos

```
// Camino 1 - 6 - 7 !OK
$id_metodologia = -1;  // metodología inválida

// Camino 1 - 2 - 3 - 4 OK
$id_metodologia = 1;  // metodología válida y
                                  // (el usuario logueado tiene permiso sobre ella
                                  // o el usuario es administrador)

// Datos correctos
$datos = array(
             'nombre'            => 'Rational Unified Process (modificado)',
             'identificador' => 'RUP (modificado)',
             'principios'  => 'Iterativo e Incremental (modificado)',
             'proceso'           => '6',
             'usuarios'          => array('3','5')
      );

// Camino 1 - 2 - 5 - 7 !OK
$id_metodologia = 1;   // Metodología válida
                       // (el usuario logueado no tiene permiso sobre ella
                      // y el usuario es no administrador)
```

CPR 1: Habilitar Metodología (camino 1)

Descripción de la Funcionalidad:

El usuario selecciona la metodología a habilitar especificando el identificador de la misma, en caso de no existir el sistema muestra un error, de lo contrario queda habilitada la metodología.

Flujo Central:

- El usuario da click en la opción del menú “Metodología/Habilitar”.
- El usuario selecciona la metodología que desea habilitar.
- Queda habilitada la metodología.

Condiciones de Ejecución:

El identificador de la metodología debe existir.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Obs.
El usuario entra un identificador de metodología válido.		El sistema muestra un mensaje de éxito de la operación.	El sistema muestra el mensaje: “Se ha habilitado la metodología”.	
	El usuario entra un identificador de metodología inválido.	El sistema muestra un mensaje de error.	El sistema muestra el mensaje: “Metodología no válida”.	

Después de realizarle un conjunto de pruebas al sistema se concluye que las pruebas constituyen un paso contundente en el ciclo de desarrollo del software que permiten verificar y revelar la calidad de un producto.

Para determinar el nivel de calidad se ha efectuado un conjunto de pruebas para comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema, aunque desde luego, los autores del presente trabajo coinciden con E.W. Dijkstra en que "El testing puede probar la presencia de errores pero no la ausencia de ellos". Lograr un software 100% libre de errores es un poco utópico aun más cuando se

trata de sistemas complejos. De hecho, uno de los problemas más difíciles con la prueba es saber cuándo parar, debido en gran medida a la variedad de pruebas que pueden aplicarse a un sistema.

El capítulo se centró en las pruebas unitarias, sin descartar las pruebas de interfaz que son necesarias como paso inicial para las pruebas de unidad debido a que verifican el flujo de la información. Se creó una clase con funciones para conducir las pruebas, se diseñaron los juegos de datos y se anotaron y evaluaron los resultados obtenidos. Es importante destacar que las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto.

CONCLUSIONES

El proceso de gestionar una metodología de desarrollo supone un gran esfuerzo para los ingenieros de software dado lo abarcador y en ocasiones abrumador que puede resultar esta tarea debido a los grandes volúmenes de información, dígase artefactos, roles, fases, disciplinas, etc., que deben manejar, no obstante es un proceso imprescindible y definitorio para el desarrollo de cualquier software. De contar con una herramienta que facilitara todo el proceso, este no resultaría tan engorroso, de ello ha tratado la implementación del Habilitador Metodológico, una herramienta web para la gestión de metodologías de desarrollo de software, para cuya implementación:

Se realizó un estudio del arte de las tecnologías actuales para el desarrollo de sistemas web, enfatizando en las tecnologías y herramientas libres, de lo que se concluyó utilizar para la implementación del sistema, como lenguaje de programación PHP 5 en su versión 5.2, con el framework CodeIgniter versión 1.7.1, sobre el entorno de desarrollo Netbeans 6.5 para PHP y utilizando como SGBD MySQL en su versión 5.0.

Se realizó una valoración crítica del diseño propuesto por el analista, que permitió adecuar esta propuesta a la implementación; se llevó a cabo un análisis a un conjunto de librerías y clases previamente implementadas a reutilizar; se analizaron los algoritmos de mayor complejidad presentes en el desarrollo del sistema; finalmente se documentaron y describieron las nuevas clases y operaciones necesarias.

Para validar la solución propuesta se condujeron un conjunto de pruebas, enfocándose fundamentalmente en las pruebas unitarias, para lo cual se creó una clase con funciones para conducir las pruebas, se diseñaron los juegos de datos y se anotaron y evaluaron los resultados obtenidos, las pruebas que constan en el presente documento son las que se practicaron a las principales funcionalidades del módulo metodología, aunque pruebas similares se aplicaron al sistema completo.

La selección de las herramientas óptimas para el desarrollo de este sistema, acompañado de un diseño que se adecúa a los retos que supone la implementación, avalado por un conjunto de pruebas que garantizan su calidad, proveen a los ingenieros de software del Habilitador Metodológico, una herramienta web para la gestión y configuración de metodologías de desarrollo de software.

RECOMENDACIONES

Con vistas a dar continuidad al desarrollo de este proyecto se recomienda:

- ✓ Ampliar la funcionalidad de exportar metodología para que permita exportar una metodología de desarrollo a otros formatos, actualmente solo a PDF.
- ✓ Ponerlo en práctica en los proyectos de la facultad, para validar que sea extensible y adaptable a diferentes metodologías de desarrollo de software.
- ✓ Crear un grupo de desarrollo para darle continuidad al proyecto de manera que pueda utilizarse en otros entornos.

Referencias Bibliográficas

1. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.*

2. *Vinny Lingham.* [En línea] [Citado el: 12 de Diciembre de 2008.] http://www.vinnylingham.com/top-20-reasons-why-web-apps-are-superior-to-desktop-apps.html.

3. Bonaparte, Ing. Ubaldo. *Paradigmas de Programación.* [En línea] 10 de Diciembre de 2004. [Citado el: 7 de Noviembre de 2008.] http://www.frt.utn.edu.ar/sistemas/paradigmas/page22.html.

4. *definición.org.* [En línea] [Citado el: 18 de Noviembre de 2008.] http://www.definicion.org/lenguaje-de-programacion.

5. S., Christian Van Der Henst. *Maestros del web.* [En línea] 23 de Abril de 2001. [Citado el: 4 de Diciembre de 2008.] http://www.maestrosdelweb.com/editorial/aspintro.

6. *El lenguaje Java.* [En línea] [Citado el: 9 de Diciembre de 2008.] http://www.dcc.uchile.cl/~lmateu/Java/Apuntes/java.htm.

7. *El lenguaje Java.* [En línea] [Citado el: 29 de Diciembre de 2008.] http://eddi.ith.mx/Curso/Contenido/java.htm.

8. *Charla "Introducción al lenguaje Perl".* [En línea] [Citado el: 6 de Diciembre de 2008.] http://www.gnuservers.com.ar/charlas/perl/index.html.

9. *Introducción al Lenguaje PERL.* [En línea] [Citado el: 06 de Diciembre de 2008.] http://soporte.ilce.edu.mx/tutoriales/perl/perl.html.

10. *PHP.net.* [En línea] [Citado el: 21 de Diciembre de 2008.] http://www.php.net/manual/es/intro-whatcando.php.

11. Luciano. *Entornos de Desarrollo Integrado para Java.* [En línea] [Citado el: 19 de Diciembre de 2008.] http://luauf.com/2008/05/13/entornos-de-desarrollo-integrado-para-java.

12. *Zend.* [En línea] [Citado el: 4 de Enero de 2009.] http://www.zend.com/en/products/studio/features.

13. *Lycos.* [En línea] [Citado el: 20 de Diciembre de 2008.] http://www.tripod.lycos.es/support/glossary/D.

14. *netbeans.org.* [En línea] [Citado el: 14 de Enero de 2009.] http://www.netbeans.org.

15. *Seguridad en Internet.* [En línea] [Citado el: 17 de Enero de 2008.] http://www.ecienciaytecnologia.gob.mx/wb2/eMex/eMex_Glosario_de_terminos_Seguridad?page=35.

16. *MySQL.* [En línea] [Citado el: 23 de Enero de 2009.]
http://dev.mysql.com/doc/refman/5.0/es/index.html.

17. Lupetti, Antonio. *Antonio Lupetti Blog.* [En línea] [Citado el: 19 de Noviembre de 2008.]
http://woork.blogspot.com/2008/11/20-great-php-framework-for-developers.html.

Bibliografía

Bacallao Guzmán, Joannis y Salazar Gutiérrez, Yanet. 2008. *Análisis y diseño del Habilitador Metodológico para la gestión de la metodología a utilizar en el desarrollo de Software.* Caracas : s.n., 2008.

Bonaparte, Ing. Ubaldo. 2004. *Paradigmas de Programación.* [En línea] 10 de Diciembre de 2004. [Citado el: 7 de Noviembre de 2008.] http://www.frt.utn.edu.ar/sistemas/paradigmas/page22.html.

Cerami, Ethan. 2002. *Web Services Essentials.* s.l. : O'Reilly, 2002. 0-596-00224-6.

Collin, Peter. *Publishing Dictionary of Computing, 4th Edition.*

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.*

Ojeda, Francisco Charte. *Programación GNU/Linux.* s.l. : ANAYA.

Pressman, Roger S. *Ingeniería del Software, Un enfoque práctico, Parte 1.*

—. *Ingeniería del Software, Un enfoque práctico, Quinta edición.*

Suehring, Steve. *MySQL Bible.*

Vázquez, José Antonio Gallego. 2003. *Desarrollo Web con PHP y MySQL.* Madrid : ANAYA, 2003. 84-415-1525-5.

Welling, Luke y Thomson, Laura. *Programación Desarrollo Web con PHP y MySQL.* s.l. : ANAYA.

Anexo 1

Los principales veinte framewoks de PHP. (17)

1. **CodeIgniter**

CodeIgniter es un poderoso y a la vez ligero framework de PHP, diseñado para desarrolladores de PHP que necesitan un simple y elegante conjunto de herramientas para crear cualquier tipo de aplicaciones Web.

2. **CakePHP**

CakePHP es un rápido framework de desarrollo para PHP que proporciona una arquitectura extensible para desarrollar, mantener y desplegar aplicaciones.

3. **Symfony**

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web escrito en PHP 5. Proporciona una arquitectura, componentes y herramientas para que los desarrolladores construyan aplicaciones web complejas más rápidamente.

4. **Prado**

PRADO™ es un framework basado en componentes y guiado por eventos para desarrollar aplicaciones web en PHP 5. Las siglas de PRADO significan PHP Rapid Application Development Object-oriented.

5. **Qcodo**

Es un framework completamente orientado a objetos que toma lo mejor de PHP y aporta una plataforma de desarrollo de aplicaciones verdaderamente rápida. Los prototipos iniciales están listos en horas en lugar de días. Las iteraciones pueden realizarse en horas en vez de días (e incluso semanas). A medida que se van concluyendo las iteraciones del proyecto, el framework permite a los desarrolladores obtener prototipos del próximo nivel en lo que madura la aplicación.

6. **Zend Framework**

Zend Framework se centra en el desarrollo de aplicaciones más robustas, confiables y modernas para la Web 2.0 & los servicios web, consumiendo una amplia gama de APIs disponibles de los principales proveedores Google, Amazon, Yahoo!, Flickr, StrikeIron y ProgrammableWeb.

7. **Akelos**

El Framework de PHP Akelos es una plataforma de desarrollo de aplicaciones web basada en el patrón de diseño MVC (Modelo – Vista - Controlador), que se enfoca en buenas prácticas que permiten programar Vistas que emplean AJAX de forma fácil, controlar las peticiones y las respuestas a través de las Controladoras, manejar aplicaciones estandarizadas internacionalmente, modelos de comunicación y de base de datos utilizando convenciones

sencillas.

8. **Maintainable**

El Framework de PHP Maintainable se desarrolló originalmente para proyectos de la empresa a la que pertenece, luego atendiendo a las solicitudes de los clientes su código fuente fue liberado. Se diseñó para el desarrollo de pequeñas a medianas aplicaciones.

9. **evoCore**

evoCore es el framework en el corazón de la evolución del blog. Está libremente disponible para su uso bajo las licencias GNU GPL o la Mozilla MPL.

10. **Stratos**

El framework Stratos es un framework para el desarrollo de aplicaciones web de código abierto, orientado a objetos que facilita el desarrollo rápido de aplicaciones web bien organizadas, seguras y de fácil mantenimiento.

11. **Seagull**

Seagull es un framework maduro en su orientación a objetos para el desarrollo de aplicaciones web, de consolas o de interfaz gráfica. Bajo licencia BSD, el proyecto permite a los desarrolladores de PHP integrar y manejar el código fuente de forma sencilla y construir aplicaciones web relativamente rápido.

12. **Zoop**

El framework Zoop es inclusivo, cooperativo y contiene components integrados de otros proyectos existentes como Smarty, el framework de Javascript Prototype y un número

considerable de módulos de PEAR.

13. **php.MVC**

php.MVC implementa el patrón de diseño Modelo – Vista – Controlador que permite a los diseñadores y programadores concentrarse en sus respectivas áreas.

14. **AjaxAC**

AjaxAC es un framework de código abierto escrito en PHP, que se utiliza para desarrollar/crear/generar aplicaciones AJAX. Se liberó bajo la licencia Apache v2.0.

15. **xAjax**

xAjax es una librería de clases de código abierto escrita en PHP que permite la creación rápida de aplicaciones Ajax empleando HTML, CSS, JavaScript, y PHP.

16. **PHOCOA**

PHOCOA es un framework para el desarrollo de aplicaciones web. Su principal plan es que el desarrollo de las aplicaciones en PHP sean más fáciles, rápidas y de mejor calidad.

17. **Kohana**

Kohana es un framework para el desarrollo de aplicaciones web, escrito en PHP 5 que emplea el patrón de diseño Modelo – Vista – Controlador. Su propósito es ser seguro, ligero y fácil de utilizar.

18. **Limb**

Limb es un framework de PHP bajo la licencia de código abierto (LGPL) diseñado principalmente para que los prototipos y el desarrollo de las aplicaciones se obtengan de forma rápida. La versión actual del framework es Limb3.

19. **Solar**

Solar es un framework para el desarrollo de aplicaciones web, escrito en PHP 5. Utiliza patrones de diseños que se adaptan a la empresa, con soporte para la localización y configuración a todos los niveles.

20. **BlueShoes**

BlueShoes es un extenso framework y manejador de contenidos para el desarrollo de aplicaciones web. Escrito en PHP. BlueShoes ofrece un excelente soporte para el popular gestor de base de datos MySQL, al igual que para ORACLE y MSSQL.

Anexo 2

Veinte razones por las cuales desarrollar una aplicación web en lugar de una de escritorio:

1. Nunca se instalan

El software que depende de un navegador nunca requiere de un proceso de instalación o de espacio en disco duro. Reside en una “nube virtual” en Internet lo que significa que cuando quiera que se ejecute, siempre estará en la última versión. Con el surgimiento del Ajax se ha hecho posible desarrollar aplicaciones web similares a las de escritorio en apariencia y sin pérdida del rendimiento.

2. Las actualizaciones son insignificantes

En lugar de aplicar parches o actualizaciones para cada usuario de forma individual, los parches o actualizaciones se aplican en el servidor y cada usuario puede contar con la versión actualizada la próxima vez que ingrese al sistema.

3. Independencia

El cual es un asunto delicado para los vendedores tradicionales de software. Usuarios que adquieren versiones anteriores de un software casi siempre terminarán dependiendo de las versiones subsiguientes teniendo en cuenta además que requieren de soporte (que es costoso). Los problemas que conciernen al software del que se depende son prácticamente ilimitados, y a menudo no resulta eficiente ni para el vendedor ni para el cliente.

4. No se requieren privilegios de administración

Finalmente, un mundo donde el administrador de red en la compañía no tiene que aprobar la instalación de tu software.

5. Disponible en todas partes a todo momento

De igual manera que las personas acceden a sus cuentas de correo desde cualquier navegador, ocurre con las aplicaciones web.

6. Independencia de la plataforma

Esto abre a los vendedores de software un mercado más amplio, ya no tienen que desarrollar tecnologías dependientes de alguna plataforma en específico y limitarse así su (o incurrir en costos adicionales en el desarrollo para otras plataformas). El navegador es la plataforma y por ende el consumo se incrementará en sistemas operativos como Mac OS y Linux, debido al incremento de la disponibilidad de aplicaciones web.

7. Menos conflictos de entorno

Se ha demostrado que se incurre en menos errores durante el desarrollo de aplicaciones web, dado que no depende de ninguna de las configuraciones de hardware o entorno en el sistema operativo que a habitualmente causan problemas.

8. Potencia las posibilidades sociales

Un gran número de aplicaciones web incorporan clientes para chatear y la posibilidad de compartir tu trabajo en tiempo real. Lo cual elimina la ya vetusta funcionalidad “stand-alone” que solía existir con la mayoría de las aplicaciones de escritorio instaladas. El mundo es cada vez más social – las personas quieren colaborar y trabajar online juntas – las aplicaciones web permiten porque sí.

9. Menor costo de venta

Sin cajas, manuales impresos, elevados costos de distribución, CD, canales de distribución, intermediarios, etc. Las aplicaciones web resultan ser más económicas de producir lo que repercute en un menor precio de venta.

10. Utilizable desde ordenadores no caros

Laptops de $100, ¡aquí vamos! ¿Para qué necesitas un procesador Core 2 Duo, si estás corriendo una ligera aplicación cliente? Esto abre todo un mundo de ahorro de costos tanto para las compañías como para los consumidores individuales, especialmente en el campo de las aplicaciones productivas (obviamente no los juegos).

11. A prueba de piratas

Imagínese un mundo sin pirateo de software. He aquí ese mundo, y las aplicaciones web son la solución a ese problema. ¡El siguiente problema, por favor!

12. Sin malas pagas

A las compañías de software a menudo los distribuidores deben dinero, que invariablemente aumentará con el tiempo. Con las aplicaciones web, el dinero se cobra al momento, mientras que el usuario pague ahí está el dinero.

13. Bajo costo en soporte y mantenimiento

Dado que ahora el navegador constituye la plataforma, el costo de soporte operacional y mantenimiento para los proveedores de aplicaciones web disminuirá substancialmente. Sin la necesidad de contar con costos especialistas que asistan con los problemas de instalación. También, el empleo de productos como la nube EC2 de Amazon, permitirá escalabilidad, sin un incremento proporcional en el costo.

14. Los datos de usuario se mantienen seguros en el servidor

Aunque esto no es necesariamente cierto para todas las compañías de aplicaciones web, pero utilizar proveedores como Rackspace o la nube EC2 de Amazon de seguro son trayecto vencido en convencer a los clientes de que su información está más segura ahí que en su escritorio.

15. No Virus

No instalación, no virus.

16. Bajo costo en distribución global

No más confianza a los canales. La mayoría de las compañías de software la crean o la rompen, en dependencia de su canal. Pues olvídelo, el mayor canal, ¡1 billón de usuarios conectados!

17. Precio de software más bajo para los clientes

Dado los anteriores beneficios, la aparición de productos tales como Basecamp y Synthasite que ofrecen, por mucho, mejores prestaciones que sus equivalentes en aplicaciones de escritorio.

18. Acceso a todos los beneficios de la Web (APIs, widgets, mensajería, colaboración).

Al estar conectado a la web, las aplicaciones web se integran sin mayores esfuerzos a diferentes API's etc y son mucho más configurables, que las aplicaciones tradicionales de software.

19. Y he aquí el móvil

Las aplicaciones compiladas de escritorio les va a costar mucho adaptarse a los dispositivos, las web ya están adaptadas (en la mayoría de los casos).

20. Audiencia potencial más amplia

Tomando en consideración todos los puntos anteriores prácticamente desbloquea el mercado a los vendedores, que anteriormente, por razones técnicas, les era inaccesible.

Glosario de términos

API: Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción

del.icio.us: Un servicio de gestión de marcadores sociales en la web.

Herramienta: Aplicación empleada para la construcción de otros programas o aplicaciones.

IDE: Entorno de Desarrollo Integrado

Ingeniería de software: es una disciplina que integra procesos, métodos y herramientas para el desarrollo de software de computadora.

Metodología: Conjunto de procedimientos o principios que guían particularmente el correcto desarrollo del software.

PEAR: Es un framework y sistema de distribución de compones reutilizables de PHP.

Plataforma: sistema operativo o sistemas complejos, ya sea de hardware o software, sobre el cual un programa pueda ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, lenguajes de programación y sus librerías de tiempo de ejecución. Ejemplos de plataformas son PC (Windows) y Macintosh (Mac).

Ruby On Rails: Entorno de programación (Rails) que se apoya en el lenguaje Ruby. Goza de gran popularidad para el desarrollo de aplicaciones de tipo Web 2.0.

Smaltalk: Primer sistema puro de objetos. Todo en Smalltalk es un objeto y toda la computación es desarrollada mediante mensajes que son enviados entre los objetos.

WYSIWYG: Siglas en inglés de (What You See is, What You Get): Se utiliza para indicar que lo que se ve en pantalla, es como se vería de forma impresa.