



**REPÚBLICA DE CUBA
MINISTERIO DE EDUCACIÓN SUPERIOR
UNIVERSIDAD DE GRANMA
DEPARTAMENTO DE INFORMÁTICA**



**UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS DE LA
INGENIERÍA Y APLICADAS**

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

**SISTEMA PARA LA GESTIÓN DE INFORMACIÓN DE LOS
PROCESOS DE PRODUCCIÓN Y CAPACITACIÓN DE
MEDICINA NATURAL Y TRADICIONAL EN GRANMA.**

**AUTORAS: MYRIAM JANETH LAMINGO CHINGO
ANA LUCIA ALVAREZ JACHO**

**TUTORES: ING. ERIC ISMAEL LEONARD BRIZUELA
M.Sc. YUDI CASTRO BLANCO
ING. RAMÓN OSMANY RAMÍREZ TASÉ**

GRANMA, CUBA

2013

DEDICATORIA

Anita

Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente.

Dedico esta tesis con mucho amor, cariño y admiración a mis queridos padres **MANUEL ALVAREZ Y CAROLINA JACHO** quienes con esfuerzo y sacrificio me han brindado su apoyo incondicional en las decisiones que he tomado, para poder llegar a esta instancia de mis estudios y por darme fortaleza para alcanzar tan anhelado sueño. Gracias por sus enseñanzas y consejos; pero sobre todo gracias por brindarme su amor.

Dedico a toda mi familia por estar conmigo en los buenos, malos momentos y por apoyarme siempre en todo, gracias por su apoyo incondicional.

AGRADECIMIENTO

Anita

Agradezco en primer lugar a Dios por darme un día más vida, que con su infinito amor me ha dado la sabiduría suficiente para culminar nuestra carrera universitaria, y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Expreso mi más grande y sincero agradecimiento a mis padres por todo el esfuerzo que hicieron para darme una profesión y hacer de mí una persona de bien, gracias por los sacrificios y la paciencia que demostraron todos estos años; gracias a ustedes he llegado a cumplir con una meta importante en mi vida.

Agradezco a mis tutores y a todas las personas quienes me brindaron sus conocimientos y apoyo para guiarme en el desarrollo de la presente tesis de grado desde el inicio hasta su culminación.

Agradezco a mis amigos quienes han vivido conmigo en la realización de esta tesis, ellos saben que desde lo más profundo de mi corazón les agradezco por haberme brindado todo su apoyo, colaboración, ánimo pero sobre todo por brindarme su cariño y amistad.

DEDICATORIA

Dedico esta tesis a mi hermosa mami Luz y a mi hermoso papi Raúl por ser quienes me dieron la vida y por brindarme su ayuda y apoyo incondicional en todo momento para culminar esta gran etapa de mi vida y sobre todo por ser mi mejor ejemplo de lucha.

A mis adorados hermanos por ser los mejores hermanos del mundo, por cuidarme tanto y brindarme ese verdadero amor de hermanos.

A mi amado esposo Diego por su amor y comprensión en los momentos más difíciles y por compartir junto a mí los momentos más hermosos de mi vida.

A mi estrellita fugaz Brigitte que aunque nunca la conocí, la llevare por siempre en mi corazón.

LOS AMO.

Myri

AGRADECIMIENTO

Agradezco principalmente a Dios por darme salud y vida y por darme la suficiente fortaleza para seguir luchando por mis sueños y mis metas.

A mis padres, hermanos y esposo por ser mi apoyo incondicional.

A mis tutores Ing. Ramón Ramírez, Ing. Eric Leonard por su paciencia y sus conocimientos brindados.

A quienes contribuyeron a la realización de la tesis. A los docentes de la Carrera de Ingeniería Informática de la Universidad de Granma Ing. Ibet Pascuales, Ing. Rony González, Ing. Erodís. Gracias por su apoyo.

GRACIAS A TODOS

Myri

Declaración de autoría

Declaramos que somos los únicos autores del trabajo de diploma titulado: **“Sistema para la gestión de información de los procesos de producción y capacitación de Medicina Natural y Tradicional en Granma”** y autorizamos a la Universidad de Granma para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ___ días del mes de _____ del _____

Firma de las Autoras

Myriam Janeth Lamingo Chingo

Ana Lucía Alvarez Jacho

Resumen

En el Centro Provincial de Medicina Tradicional (MNT) en Granma ubicado en la ciudad de Bayamo, se lleva a cabo la gestión de información de los procesos de producción y capacitación. Este proceso se realiza de forma compleja, debido a que se debe registrar dicha información de un modo constante para así tener el control preciso de los datos, ya sea en la dirección provincial como en cualquiera de sus entidades.

Atendiendo a estas necesidades, se desarrolla una aplicación para facilitar la gestión de los procesos de producción y capacitación en el Centro de Medicina Natural y Tradicional de forma rápida, confiable y segura. El proceso de desarrollo de la aplicación se realiza apoyándose en tecnologías libres, multiplataforma, utilizando PHP 5.3.3 como el lenguaje de programación, se implementa el patrón de arquitectura Modelo-Vista-Controlador, utilizando el framework CodeIgniter, MySQL 5.5.20 como Sistema Gestor de Bases de Datos y Netbeans7.3 como Entorno de Desarrollo Integrado.

Summary

In the Provincial Center of Traditional Medicine (MNT) in Granma located in Bayamo city, it gets to stub the step of information of the processes of production and training. This process comes true of way complex, because said information of a constant mode in order to have the precise control of the data should get registered, either in the provincial direction as in anyone of his entities.

Attending to these needs, develops an application to facilitate the step of the processes of production and training in Traditional Medicina's Center of fast, reliable and safe way. The process of development of the application comes true basing itself on free technologies, multi-platform, using PHP 5.3.3 like the programming language, takes effect of Model-View-Controller architecture, using the framework CodeIgniter, MySQL 5.5.20 as System Managing of Data Bases and Netbeans 7.3 like Integrated Development Enviroment.

Índice General

Introducción	1
Capítulo 1. Fundamentación teórica.....	5
1.1 Introducción	5
1.2 Sistemas para la gestión del proceso técnico	5
1.3 Herramientas, tecnologías y metodologías para el desarrollo de sistemas informáticos.....	6
1.4 Lenguajes de Programación	9
1.4.1 Lenguajes del lado del Cliente.....	9
1.4.2 Lenguajes del lado del Servidor.....	10
1.5 Servidor Web	14
1.5.1 El servidor Web Apache	14
1.6 Sistemas Gestores de Bases de datos (SGDB).....	15
1.7 Frameworks	16
1.8 Herramienta de desarrollo para PHP	19
1.9 Metodologías de desarrollo de software	20
Justificación de las herramientas seleccionadas	25
1.10 Conclusiones del capítulo	26
Capítulo 2 Descripción, valoración y construcción de la propuesta de solución	27
2.1 Introducción	27
2.2 Personas relacionadas con la aplicación	27
2.3 Fase de exploración.....	28
2.3.1 Historias de usuarios	28
2.3.2 Valoración de sostenibilidad	30
2.4 Fase de Planificación	32
2.4.1 Estimación de esfuerzos por Historias de Usuario	32
2.4.2 Plan de duración de las iteraciones	33
2.4.3 Plan de entrega	34
2.5 Iteraciones	36
2.5.1 Tareas	37

2.6 Producción	41
2.6.1 Diseño	41
2.6.2 Codificación	44
2.6.3 Prueba	44
2.7 Mantenimiento	47
2.8 Muerte del Proyecto	47
2.9 Conclusiones del capítulo	47
Conclusiones Generales	48
Recomendaciones.....	49
Referencias bibliográficas.....	50
Bibliografía	53
Anexos	LV

Índice de Tablas

Tabla 1.1 Comparación entre metodologías ágiles y tradicionales	21
Tabla 2.1 Personas relacionadas con la aplicación.	27
Tabla 2.2 Historia de Usuario Gestionar Cuenta de Usuario.....	29
Tabla 2.3 Historia de Usuario Gestionar Trabajador.	29
Tabla 2.4 Historia de Usuario Gestionar Entidad.	29
Tabla 2.5 Estimación de esfuerzos por Historia de usuario.	33
Tabla 2.6 Plan de duración de las iteraciones.....	34
Tabla 2.7 Módulos e Historias de Usuario.....	35
Tabla 2.8 Módulos e Iteraciones.	36
Tabla 2.9 Tareas establecidas en cada iteración.	37
Tabla 2.10 Tarea Diseño de la interfaz de Gestionar Cuenta de Usuario.	40
Tabla 2.11 Tarea Insertar Usuario.	41
Tabla 2.12 Tarea Desactivar Usuario.....	41
Tabla 2.13 Tarjeta CRC Módulo Curso	43
Tabla 2.14 Tarjeta CRC Módulo Tema	44
Tabla 2.15 Caso de prueba Diseño correcto de la interfaz para la selección de los usuarios.	45
Tabla 2.16 Caso de prueba Insertar correctamente usuario	46
Tabla 2.17 Caso de prueba error insertar usuarios.....	46

Índice de Anexos

Anexo 1 Imágenes de la solución propuesta LV
Anexo 2 Historias de usuarios LVII
Anexos 3 Tareas para realización de las Historias de Usuario LIX
Anexo 4 Diseño de la Base de Datos LXI
Anexo 5 Tarjetas CRC LXII
Anexo 6 Pruebas de aceptación realizadas al sistema.....LXV

Introducción

En la actualidad la información y las comunicaciones, son factores extremadamente claves en los procesos de producción y obtención de mejoras. Las Tecnologías de la Informática y las Comunicaciones (TIC) han demostrado ser instrumentos que pueden contribuir al logro de amplios objetivos nacionales, tanto sociales como económicos, en la medida en que los estados las incorporen a las principales políticas y programas de desarrollo de sus naciones.

La informática cumple un papel importante a la hora de organizar y gestionar la información puesto que las nuevas tecnologías impactan en la vida de las personas, sociedad, cultura, educación, trabajo, y organización, consecuencia de la evolución tecnológica de los últimos años.

Las aplicaciones desarrolladas en las computadoras facilitan realizar una buena gestión de la información y a su vez posibilitan un acceso rápido y fácil, esto permite tomar una decisión más acertada utilizando la tecnología informática actual (Aguilar, 2013).

A nivel mundial va tomando auge la Informática y como parte de ella la producción de software; incluyendo Cuba; país en donde las entidades han ido descubriendo que requieren actualizarse al ritmo de los avances tecnológicos y es necesario que las mismas cuenten con un sistema de gestión de información que permita organizar, procesar y poner en uso la información de la organización (tanto de origen externo como interno) y utilizar adecuadamente los recursos de las empresas.

En la provincia Granma el Centro de la Medicina Natural y Tradicional lleva a cabo el control y procesamiento de la información referente a los procesos de producción de fármacos y capacitación del personal de forma manual, lo que ocasiona pérdidas de información para la entidad y en ocasiones ilegibilidad en los modelos y formularios impresos. Además se realizan llamadas telefónicas y viajes a los municipios de la provincia para la recogida de los datos de los centros productores y otras entidades subordinadas, generando como resultado la necesidad de un sistema informático que permita un mejor control, centralización y actualización de la información de dichos procesos (Guerra J y Guerra D, 2011).

Por lo antes planteado se decidió la creación de un sistema informático para disminuir el tiempo y esfuerzo, garantizar un fácil procesamiento de la información y obtención dinámica de los reportes facilitando la toma de decisiones y las acciones correctivas necesarias de forma inmediata. Constituyendo esta, la meta del presente trabajo de investigación.

Dadas las ineficiencias y atendiendo a las necesidades que presenta el Centro de Medicina Natural y Tradicional se define el siguiente **problema científico**: ¿Cómo favorecer la gestión de la información de los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma?

Teniendo como **objeto de investigación**: La gestión de información de los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en Granma.

Enmarcado en el siguiente **campo de acción**: sistema de gestión de información de los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.

La presente investigación tiene como **idea a defender**: el desarrollo de un sistema informático facilitará un mejor control de la información referente a los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.

Con el fin de darle solución a la situación problemática se plantea como **objetivo general** de la investigación: Desarrollar un sistema informático para la gestión de la información de los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.

Para darle cumplimiento al objetivo general anteriormente planteado, se trazaron los siguientes **objetivos específicos**:

- Caracterizar el proceso de gestión de la información de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.
- Implementar un sistema informático para la gestión de la información de los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.

Para dar cumplimiento a los objetivos específicos se plantean las siguientes **tareas de investigación**:

- Revisar bibliografía científica referente al proceso de producción y capacitación.
- Analizar los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.
- Analizar las tendencias actuales de las metodologías, tecnologías y herramientas a utilizar en el desarrollo de sistemas informáticos.
- Diseñar y normalizar la base de datos que mantenga la información persistente.
- Implementar un sistema informático para la gestión de la información referente a los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.
- Desplegar el sistema propuesto en la presente investigación en el Centro de Medicina Natural y Tradicional en Granma.

Para dar cumplimiento a las tareas anteriores se utilizaron los siguientes métodos y técnicas:

Métodos teóricos

- **Análisis y Síntesis:** Se empleó para la recopilación y el procesamiento de la información y arribar a las conclusiones de la investigación, la obtención de conocimiento y resumir la información a procesar.
- **Histórico-lógico:** Para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
- **Modelación Sistémica:** Para mostrar la forma en que el sistema tiene que funcionar y para estudiar cómo se combinan los distintos componentes a fin de producir algún resultado.

Método Empírico

- **Observación:** se utilizó para acercarse al trabajo de los clientes potenciales del subsistema y por tanto a los procesos que se necesitan informatizar.

Técnica

- **Entrevista:** Es empleada con el personal que estará interactuando con el sistema, con el fin de obtener conocimientos más específicos de cómo funciona el proceso de gestión de información, en el Centro de MNT en Granma, quiénes lo manejan y quiénes son los encargados de los procesos, permitiendo definir el problema a resolver y establecer el objeto de estudio.

El documento está estructurado en introducción, dos capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, y anexos.

El **capítulo 1:** se realiza un estudio de los temas relacionados con el objeto de estudio y el campo de acción de la investigación, así como un estudio bibliográfico actualizado sobre las principales tendencias y tecnologías actuales, incluyendo las herramientas usadas en el desarrollo del sistema informático.

El **capítulo 2:** se abordan las principales características del sistema, se describe el objeto de estudio y se exponen todas las fases pertenecientes a la metodología de desarrollo Extreme Programming utilizada para la implementación de la aplicación.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En el presente capítulo se hace un análisis del objeto de estudio y el campo de acción de la investigación para la selección de los procesos a informatizar. Se abordan las tendencias y tecnologías actuales en cuanto a lenguajes de programación Web, servicios Web, diseño de interfaz, sistemas gestores de bases de datos y metodologías para el desarrollo de sistemas informáticos; haciendo énfasis en las que se usan en el desarrollo de aplicaciones web.

1.2 Sistemas para la gestión del proceso técnico

Cuando se define el término gestionar se habla de coordinar todos los recursos disponibles para conseguir determinados objetivos. Es el proceso que desarrolla actividades productivas con el fin de generar rendimientos de los factores que en él intervienen. Por lo que tanto en Cuba como en el resto del mundo se gestiona información de disímiles maneras (Corona, 2010).

Software de empresas internacionales:

- ✓ **QTraining:** Es una poderosa pero sencilla herramienta para evaluar el desempeño del personal, detectar necesidades de capacitación y administrar los cursos de capacitación de la empresa. Permite evaluar el impacto de las capacitaciones llevadas a cabo y relaciona los objetivos de desempeño con los cursos de capacitación (Wilsoft, 2011).
- ✓ **MasAdmin Farmacias:** Está basado en el gran sector productivo como lo es el farmacéutico, es multiusuario, multiempresa y multialmacén. Permite el control total de listas de precios, es un poderoso y versátil punto de venta compatible con lectores de códigos de barras y le permite registrar la cédula y el nombre del doctor quien expide las recetas; posee además la posibilidad de crear una gran diversidad de reportes (MasAdmin, 2009).

Software de empresas de Cuba:

- ✓ **PresCaP:** Herramienta integradora para la gestión de la capacitación en la corporación CIMEX S.A en el marco de la norma cubana **NC 3002:2007**, con el objetivo de mejorar las insuficiencias existen en la planeación y control de

las acciones de capacitación a nivel de entidades en dicha corporación (Guerra J y Guerra D, 2011).

- ✓ **CEMARN:** Sistema de información para la gestión del conocimiento en la Universidad de Pinar del Río, con el objetivo de posibilitar un mejor desempeño informacional en la actividad investigativa del colectivo de investigadores de dicho centro de estudio (Guerra J y Guerra D, 2011).
- ✓ **SIGIMNT:** Sistema de gestión para el control de los procesos de producción y docencia de la Medicina Natural y Tradicional en Granma, con el objetivo de disminuir el tiempo y esfuerzo, garantizar un fácil procesamiento de la información y obtención dinámica de los reportes facilitando la toma de decisiones de la empresa de MNT (Guerra J y Guerra D, 2011).

Todas estas aplicaciones son desarrolladas para Windows y por lo tanto solo se podrán ejecutar desde estaciones de trabajo con este sistema operativo instalado, lo que limita grandemente la utilización de otros sistemas operativos. La mayoría, son aplicaciones de escritorios desarrolladas mayormente con software propietarios, por lo que impide el acceso o publicación de un gran número de informaciones útiles para el personal no especializado en los procesos de producción y capacitación, son mayormente orientados a locales de trabajos con especialistas en la materia.

En el caso específico de SIGIMNT es un software que no tiene en cuenta la producción de medicamentos, además de no estar implementado usando buenas prácticas de programación, por otro lado con el diseño de interfaz después de desplegado el sistema en la empresa surgieron inconformidades pues la forma en que está diseñado fatiga a los usuarios.

1.3 Herramientas, tecnologías y metodologías para el desarrollo de sistemas informáticos

Aplicaciones Web

Una Aplicación Web es un conjunto de páginas Web enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de información. Permiten modificar la lógica del negocio a través de un navegador (Cinca, 2008).

Una página web contiene elementos que permiten una comunicación activa entre el usuario y la información logrando el usuario acceso a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones.

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, que soportan por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva (Betancourt, 2010).

Entre las principales comodidades que una aplicación Web pueden traer se encuentran las siguientes:

- **Compatibilidad multiplataforma.** Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos (Colombo B y Demetrescu C, 2003).
- **Actualización.** Las aplicaciones basadas en web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones) (Oliver, 2000).
- **Inmediatez de acceso.** Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y listos para trabajar sin importar cuál es su configuración o su hardware.
- **Facilidad de prueba.** Finalmente no habrá más obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones antes de cargar su tarjeta de crédito. Actualmente, especialmente cuando se habla de software costoso, hay todavía una gran cantidad de funcionalidades y pequeños detalles que no pueden ser totalmente probados descubiertos antes de comprometer dinero en alguna compra total.

- **Menos requerimientos de memoria.** Las aplicaciones basadas en web tienen mucha menos demanda de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y ejecutarse en los servidores del proveedor, esas aplicaciones web usan en la mayoría de los casos la memoria de las computadoras en que se ejecutan, dejando más espacio para la ejecución de múltiples aplicaciones al mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs.** Las aplicaciones basadas web, deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos. Esta es la razón por la cual estas aplicaciones deberían tener mucho menos bugs que el software de escritorio descargable tradicional (Zelkovitz M, Shaw A y Gannon J, 2009).
- **Precio.** Las aplicaciones basadas en web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Múltiples usuarios concurrentes.** Las aplicaciones basadas en web puedan realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. Las compañías de conferencia web y colaboración online están involucradas en algunas transformaciones claves y los usuarios necesitan explorar que significa realmente trabajar efectivamente y co-editar documentos juntos (Brusilovsky P, Gilles G, Frasson C y VanLehn K, 2007).
- Usando la Web, se tiene acceso a millones de páginas de información, la exploración se realiza por medio de un software especial denominado browser o explorador. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero (Prentzas J, Hatzilygeroudis I y Garofalakis J, 2002).

1.4 Lenguajes de Programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes (Arias, 2008).

En la programación de la web se habla de dos tipos de lenguajes: los lenguajes del lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente, que son los encargados de aportar dinamismo a la aplicación en los navegadores (Furgeta, 2002).

Entre los lenguajes del lado del servidor se pueden encontrar entre los más sobresalientes al *Active Server Pages* (ASP), *Personal Home Pages* (PHP), *Java Server Pages* (JSP), *Practical Extraction and Report Language* (PERL). Del lado del cliente se encuentra el HTML, Java y JavaScript los cuales son simplemente incluidos en el código HTML (De la Torre, 2006).

1.4.1 Lenguajes del lado del Cliente

***Hypertext Markup Language* (HTML)**

HTML es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a *HyperText Markup Language*, es decir, Lenguaje de Marcas de Hipertexto, en donde hipertexto hace referencia a la capacidad del lenguaje para la utilización de los conocidos hipervínculos. EL HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos, fotografías, animaciones, etc.

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto, en él se crea a partir de etiquetas, también llamadas *tags*, que permiten interconectar diversos conceptos y formatos (Wed, 2013).

Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomó dos herramientas preexistentes: El concepto de Hipertexto (conocido también como link

o ancla) el cual permite conectar dos elementos entre sí y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda (Ruiz, 2006).

Java Script

Es un lenguaje interpretado, multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento, usado para el desarrollo de aplicaciones cliente-servidor en páginas (Málaga, 2008).

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. *Javascript* es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, *Javascript* pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente (Alvarez, 2001).

1.4.2 Lenguajes del lado del Servidor

PHP

PHP es el acrónimo de *Hipertext Preprocesor*. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores (García, 2006).

Una vez que ya se conoce el concepto de lenguaje de programación de scripts del lado del servidor se puede hablar de PHP. PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad. Cualquiera puede descargar a través de la página principal de PHP, www.php.net y de manera gratuita, un módulo que hace que nuestro servidor web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

PHP, en el caso de estar montado sobre un servidor Linux u Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP (Eloi, 2009).

Respecto a la seguridad, es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores (Cobo Á, Gómez P, Pérez D y Rocha R, 2010).

Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes como: MySQL, MSSQL, Oracle, Informix, y ODB. Incluye funciones para el envío de correo electrónico, subida de archivos, crear dinámicamente en el servidor, imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

Características de PHP

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de bases de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo-Vista-Controlador.

- Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principio de los años 90 que requiere un proceso de compilación. El código compilado puede ser integrado en la página web para ser ejecutado por el cliente (Málaga, 2008).

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la web en un elemento más interesante y útil. Java no es lo mismo que *javascript*, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.

Con Java se pueden crear dos tipos de programas:

Applets: programas que se integran en las páginas web y que, residiendo en el servidor, son ejecutados por el cliente. La ejecución necesita de la interpretación del código compilado por el software cliente (Askallorg, 2007).

Aplicaciones: Programas autónomos que se pueden ejecutar en cualquier equipo. En este último caso puede optarse por generar código compilado similar al de los *applets* y que para su ejecución necesita de un intérprete o código compilado ejecutable directamente como en cualquier otro lenguaje de programación.

En el caso de *applets*, el código fuente no se aloja directamente en el documento HTML, si no al que se lo añade es un código binario resultado de la compilación, el denominado JBC (*Java Byte Code*). Esto permite proteger el código fuente, aunque hasta cierto punto, ya que las particularidades de este código compilado hacen que sea factible el proceso inverso, es decir, la descompilación, recuperar el código fuente a partir del compilado. En la propia Internet puede encontrarse programas capaces de hacerlo. La razón de todo esto está en el hecho de que para conseguir la portabilidad de los programas el código compilado es un código que se encuentra entre el código fuente y un código objeto fuertemente dependiente de una plataforma. Es por ello que se suele decir de Java que es un lenguaje que combina

la flexibilidad de los lenguajes interpretados y el poder de los compilados (JBC, 2009).

1.5 Servidor Web

Es un programa que sirve datos en forma de Páginas Web, hipertextos o páginas HTML (*HyperText Markup Language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo Http (De la Torre, 2006).

El servidor responde al cliente enviando el código HTML de la página; el navegador cuando recibe el código, lo interpreta y lo muestra en pantalla. El Cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. El servidor se encarga de transferir el código de la página sin llevar a cabo ninguna interpretación de la misma (Vásquez, 2008).

1.5.1 El servidor Web Apache

Apache es el servidor Web por excelencia, su configurabilidad, robustez y estabilidad hacen que millones de servidores reiteren la confianza en sus servicios. Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una simple lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores.

Apache es un Servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular.

Este servidor web es gratuito, y de código abierto, es decir corre sobre cualquier plataforma. Apache es una muestra, al igual que el Sistema Operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar (Aldana, 2009).

1.6 Sistemas Gestores de Bases de datos (SGDB)

Un Sistema de Gestión de Bases de Datos (SGBD) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación (Cobo y Gómez, 2005).

Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.

Los SGBD permiten mantener la seguridad mediante el establecimiento de claves, para identificar al personal autorizado a utilizar la base de datos. Mejoran la accesibilidad a los datos proporcionando lenguajes de consultas o generadores de informes, que permiten al usuario establecer cualquier tipo de consulta sobre los datos, sin necesidad de un programador que escriba una nueva aplicación para realizar tal tarea (Bertino E y Martino A, 1995).

A pesar de las muchas ventajas que ofrecen los SGBD, también tienen muchas desventajas, por ejemplo, son un conjunto de programas muy complejos con una gran funcionalidad. Son muy extensos y requieren gran cantidad de espacio en disco y en memoria para trabajar de forma eficiente. Los SGBD están escritos para ser generales y útiles en muchas aplicaciones, lo que generalmente conlleva a una pérdida de rapidez en el funcionamiento de estas últimas (Angel, 2010).

Entre los SGBD más usados se pueden mencionar el Oracle, MySQL, *Microsoft SQL Server*, PostgreSQL, entre otros.

PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de *Postgres*, desarrollado en la Universidad de California.

Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multiversión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación (Cabrera, 2010).

Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (*constraints*) y los disparadores (*triggers*). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales.

Posee una integridad referencial e interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG; PYTHON y RUBY. Funciona en todos los sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico (Lockhart, 1996).

MySQL

Es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas.

Soporta clientes en C, C++, Eiffel, Java, Perl, PHP, Python y Tcl. Trabaja en diferentes plataformas además; de soportar múltiples idiomas, completo y optimizado uso del SQL y contiene un MyODBC. Está disponible en diferentes plataformas, es gratis, se puede disponer del código fuente (Petckvic, 2005).

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones (Cabrera, 2010).

1.7 Frameworks

Un *framework* es un diseño reutilizable del todo o la parte de un sistema/software descrito por varias jerarquías de herencia de clases (generalmente algunas

abstractas) y las distintas colaboraciones que se establecen entre las instancias de estas.

El uso de un framework minimiza el tiempo de desarrollo de un sistema, ya que proporciona sub-sistemas que funcionan, lo que reduce su codificación y acelera su puesta en marcha. El framework reduce los riesgos de desarrollo del software, proporcionando una base fiable suficientemente probada y una arquitectura consistente entre aplicaciones.

Al usar frameworks, todas las aplicaciones generadas comparten una arquitectura común, facilitándose su comprensión, mantenimiento y soporte. Cualquier programador al trabajar con un framework, no deberá invertir gran parte de su tiempo en buscar las clases necesarias, interconectarlas, ni descubrir los métodos que estas contienen. Los frameworks ocultan toda esa complejidad, dando un alto nivel de abstracción y facilitando el desarrollo del software, permitiendo pasar más tiempo identificando requerimientos para el mismo que tratando con los detalles de bajo nivel (Crespo P y Garcia P, 2000).

Zend Framework

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones Web y servicios Web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "*use-at-will*" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de *Zend Framework* conforman un potente y extensible framework de aplicaciones Web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de bases de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como *Zend_Auth* y *Zend_Acl*, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes. También existen

componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los *Web services* más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de *Zend Framework* que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida.

El principal patrocinador del proyecto *Zend Framework* es *Zend Technologies*, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios Web y otras tecnologías que desean poner a disposición de los desarrolladores de *Zend Framework* (Zuares, 2009).

CodeIgniter

Es un conjunto de herramientas para construir aplicaciones web desarrolladas en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

Como cualquier otro *framework*, *CodeIgniter* contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debe seguir para obtener provecho de la aplicación (Alvarez, 2001).

Utiliza el acercamiento al MVC permitiendo una buena separación entre lógica y presentación, lo que ayuda a programar de manera más ordenada. Contiene también diversas herramientas que le proporcionan a las aplicaciones un mayor grado de versatilidad y seguridad (Carrasco, 2009).

Características generales de CodeIgniter

Algunos de los puntos más interesantes sobre este framework, sobre todo en comparación con otros productos similares, son los siguientes:

- ✓ **Versatilidad:** *CodeIgniter* es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo se tiene un acceso por FTP para enviar los archivos al servidor y donde no se tiene acceso a su configuración.
- ✓ **Compatibilidad:** *CodeIgniter*, es compatible con la versión PHP 4 y PHP 5, lo que hace que se pueda utilizar en cualquier servidor.

- ✓ **Facilidad de instalación:** Su configuración se realiza con la edición de un archivo, donde se escribe el acceso a la base de datos. Durante la configuración no se necesita el acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- ✓ **Flexibilidad:** CodeIgniter es menos rígido que otros *frameworks*. Define una manera de trabajar específica. Algunos módulos como el uso de plantillas son totalmente opcionales.
- ✓ **Ligereza:** El núcleo de CodeIgniter es ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional.
- ✓ **Documentación tutorializada:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial.

Symfony

Symfony es un *framework* completo diseñado para optimizar el desarrollo de las aplicaciones Web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web.

Symfony está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. *Symfony* es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows (Labs, 2012).

1.8 Herramienta de desarrollo para PHP

NetBeans

NetBeans es un entorno de desarrollo muy completo y profesional. Contiene muchas funcionalidades, para distintos tipos de aplicaciones y facilita la programación, la

prueba y la depuración de las aplicaciones que se desarrollan, también incorpora un editor propio.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Leonard, 2011).

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

1.9 Metodologías de desarrollo de software

El concepto de metodología, dentro de la Ingeniería del Software es uno de los más oscuros y que más confusión produce tanto en estudiantes como en profesionales involucrados en procesos de desarrollo de software.

Tanto es así, que en muchos proyectos de desarrollo (no todos, por supuesto), la aplicación de una metodología brilla por su ausencia, lo que hace al concepto casi desconocido.

Las Metodologías de Desarrollo de Software se pueden dividir en dos grupos: metodologías ágiles y las metodologías tradicionales (no ágiles). Las metodologías ágiles se centran en el uso de mínima documentación en el momento de desarrollar los paquetes y buscan ser ágiles en cuanto a responder adecuadamente a los cambios de requerimientos. Mientras que las metodologías tradicionales (no ágiles) se encargan de dividir el proceso de desarrollo en etapas, de una manera secuencial, son más un resultado de una urgencia por dotar al desarrollo de software de orden para poder completar los objetivos deseados, las cuales proveen de una exagerada descripción del modelo del software pero no indican cómo deben llevarse

a cabo las tareas del proyecto, por estar enfocadas en procesos, y no en personas (Canós J, Letelier P y Penadés C, 2003).

En la tabla siguiente se muestra una comparación entre las metodologías ágiles y las convencionales.

Tabla 1.1 Comparación entre metodologías ágiles y tradicionales

Metodología Ágil	Metodología No Ágil (Tradicional)
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambio
Pocos artefactos	Más artefactos
Pocos roles	Más roles
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Menos énfasis en la arquitectura del software	La arquitectura de software es esencial y se expresa mediante modelos

Todo desarrollo de software requiere un conjunto de actividades complejas, las cuales al realizarse sin tener una metodología de desarrollo puede producir insatisfacción en los clientes.

Para guiar el proceso de desarrollo de software existen diferentes metodologías, las más reconocidas mundialmente son:

☞ *Rational Unified Process (RUP).*

☞ *Extreme Programming (XP).*

RUP (RATIONAL UNIFIED PROCESS)

La metodología Proceso Unificado de Desarrollo de Software (RUP) se basa en la orientación a objetos, el desarrollo iterativo y el modelamiento visual usando el Lenguaje Unificado de Modelación (UML) para la ingeniería de sistemas y de software, lo que permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Esta metodología proporciona al equipo del proyecto procedimientos y herramientas promoviendo las siguientes prácticas (Jacobson, Booch y Rumbaugh, 2000; Prentzas J, et al., 2002):

- Gestión de los Requisitos.
- Desarrollos iterativos.
- Uso de arquitecturas basadas en componentes.
- Desarrollo Visual del Software con UML.
- Verificación continua de la calidad del software.
- Gestión de los Cambios.

Según la metodología RUP el ciclo de vida de un proyecto se divide en las siguientes fases:

- Fase de concepción.
- Fase de elaboración.
- Fase de construcción.
- Fase de transición

A pesar de ser una metodología desarrollada directamente para el trabajo con clases y objetos brinda amplias posibilidades para el manejo eficiente del tiempo de diseño e implementación utilizando las extensiones para aplicaciones Web (Furgeta, 2002).

Programación Extrema (XP)

Es una metodología ágil para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo. La Programación Extrema es una metodología ágil centrada en potenciar las relaciones

interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y estableciendo un buen clima de trabajo. Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca simplificar las soluciones implementadas y coraje para los múltiples cambios. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo (Guerrero, 1999).

La metodología de XP se basa en:

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelanta en algo hacia el futuro, se puede hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- **Re fabricación:** Se desarrolla en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento (Díaz, 2009).

Características generales de XP

- Los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos.
- Los diseños de software se mantienen sencillos y libres de complejidad o pretensiones excesivas.
- Se obtiene retroalimentación de usuarios y clientes desde el primer día gracias a las pruebas.
- El software es liberado en entregas frecuentes tan pronto como sea posible.
- Los cambios se implementan rápidamente tal y como fueron sugeridos.
- Las metas en características, tiempo y costos son reajustadas permanentemente en función del avance real obtenido.

XP consta de seis fases:

Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Planificación de la Entrega (Release): Es la comunicación frecuente entre clientes y programadores, donde las personas del negocio definen el ámbito, la prioridad de lo que debe hacerse primero, la composición de las versiones, la fecha de versiones, estimación, consecuencias, procesos y programación detallada.

Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

Producción: La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación.

Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo (Joskowicz, 2008).

Justificación de las herramientas seleccionadas

La selección de las tecnologías a utilizar para el desarrollo del sistema informático se basó fundamentalmente en la utilidad de las mismas, en la plataforma disponible y en las exigencias del cliente para mantener la mayor integración posible entre las mismas. En el desarrollo del sistema propuesto se utiliza las siguientes herramientas.

- PHP 5.3.3 como lenguaje de lado servidor, ya que fue diseñado específicamente para el desarrollo de páginas dinámicas, permite incluir el código HTML con facilidad e incluye un sinnúmero de bibliotecas para el soporte de aplicaciones Web sin costo alguno (e-mail, generación de ficheros PDF, etc.).
- El lenguaje HTML, por ser de su tipo el que predomina para la creación y publicación de páginas Web y al que se le pueden añadir numerosas funcionalidades mediante las CCS y el código JavaScript, obteniéndose páginas Web rápidas y sencillas.
- MySQL 5.5.20 como gestor de bases de datos, por ser multiplataforma, libre, robusto y rápido, entre otras funcionalidades.

- Apache 2.2.21 como servidor Web, pues ofrece un amplio soporte para PHP, es una tecnología de código abierto, multiplataforma, extensible y se considera uno de los más utilizados actualmente.
- El framework CodeIgniter, ya que implementa buenas técnicas y patrones de diseño para la creación de aplicaciones Web.
- XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, y se define especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes.

1.10 Conclusiones del capítulo

Al finalizar el capítulo se ha expuesto todo lo referente a la base teórica, que fundamenta la presente investigación, se realiza un estudio del estado del arte de las soluciones informáticas para el control de los procesos de producción y capacitación del Centro de MNT en Granma, concluyéndose que ninguna de las existentes permite gestionar las actividades asociadas específicamente a un proceso de despliegue.

Se realiza un análisis profundo de las metodologías de desarrollo y de las tendencias actuales de las tecnologías. Concluyendo que el desarrollo del sistema será guiado por la metodología XP, ya que cubre el ciclo de vida de una aplicación web; y se valora el porqué de la elección referente al lenguaje de programación a usar, el sistema gestor de bases de datos, la herramienta de desarrollo, y el servidor Web.

Capítulo 2 Descripción, valoración y construcción de la propuesta de solución

2.1 Introducción

Este capítulo está enmarcado en las fases de exploración, planificación, iteraciones, producción, mantenimiento y muerte del proyecto, definidas en la metodología escogida. Además, se describe la construcción de la propuesta con el desarrollo de las historias de usuario a través de iteraciones, se definen tareas y principios de diseño para su implementación. Se exponen pruebas de aceptación para garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

2.2 Personas relacionadas con la aplicación

Se denomina persona relacionada con la aplicación aquella que interactúa e intercambia con la aplicación y obtiene un resultado de uno o varios procesos que se ejecutan en el mismo (Pressman, 2002). En la siguiente tabla se detalla una breve descripción sobre las actividades que puede realizar una persona con la aplicación.

Tabla 2.1 Personas relacionadas con la aplicación.

Persona Relacionada con el Sistema	Justificación
Administrador	Persona(s) facultada(s) para la gestión del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios en la aplicación. Cuenta además con todos los privilegios.
Capacitador	Persona(s) facultada(s) para la gestión de la capacitación. Es el encargado de administrar los diferentes temas, cursos bibliografía y docentes en la aplicación.
Especialista de Producción	Persona(s) facultada(s) para la gestión de la producción. Es el encargado de administrar

	las entidades, plan producción, extra plan y productos en la aplicación.
Usuario	Representa la(s) persona(s) que solo pueden generar los reportes de la aplicación.

2.3 Fase de exploración

Es la fase en la que se define el alcance general del proyecto. El cliente plantea a grandes rasgos las historias de usuario que son de interés para el desarrollo del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura de la aplicación construyendo un modelo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con las tecnologías y herramientas a utilizar (Pressman, 2002).

2.3.1 Historias de usuarios

Las historias de usuario (HU) son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que la aplicación debe realizar, estas deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Joskowicz, 2008). A continuación se muestran algunas historias de usuarios definidas en el proceso de desarrollo de la aplicación, se pueden consultar otras en el [Anexo 2](#).

Tabla 2.2 Historia de Usuario Gestionar Cuenta de Usuario.

Historia de Usuario	
Número: H1	Usuario: Administrador.
Nombre de Historia: Gestionar Cuenta de Usuario	
Prioridad en Negocio: Alta	Puntos Estimados: 0.3
Riesgos en Desarrollo: Alta	Iteración Asignada: 1
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se podrá crear, modificar y eliminar las cuentas de usuario en el sistema.	
Observaciones:	

Tabla 2.3 Historia de Usuario Gestionar Trabajador.

Historia de Usuario	
Número: H2	Usuario: Administrador, Capacitador.
Nombre: Gestionar Trabajador	
Prioridad en Negocio: Alta	Puntos Estimados: 0.3
Riesgos en Desarrollo: Alta	Iteración Asignada: 1
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se brinda la posibilidad de insertar, modificar, eliminar y buscar un trabajador.	
Observaciones:	

Tabla 2.4 Historia de Usuario Gestionar Entidad.

Historia de Usuario	
Número: H3	Usuario: Administrador.
Nombre: Gestionar Entidad	
Prioridad en Negocio: Alta	Puntos Estimados: 0.3
Riesgos en Desarrollo: Alta	Iteración Asignada: 1
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	

Descripción: Se brinda la posibilidad de insertar, modificar, eliminar y buscar una entidad por varias características.
--

Observaciones:

2.3.2 Valoración de sostenibilidad

Con la elaboración del sistema informático se puede incidir de forma positiva y negativa sobre los usuarios finales, es por eso que se hace importante realizar una valoración de su impacto en las dimensiones administrativas, socio-humanista, ambiental y tecnológico, para saber hasta qué punto es sostenible la aplicación informática elaborada.

Dimensión Administrativa

Para el desarrollo del Sistema para la Gestión de Información de Medicina Natural y Tradicional en Granma no habrá gastos por mano de obra, puesto a que forma parte de un trabajo de diploma, además para toda la fase de diseño, implementación, y prueba se utilizará equipamiento ya existente en la Universidad de Granma.

Al implantarlo no se incurrirá en gastos adicionales pues existe el equipamiento necesario ya que es utilizado en otras actividades, por lo que tampoco se incrementará el consumo de electricidad.

La programación del sistema informático se realizará en PHP y JavaScript, como servidor de aplicaciones Web Apache y el sistema gestor de base de datos MySQL los cuales son gratuitos, por lo que no se incurrirá en gastos de licencia o adquisición.

Dimensión Socio-humanista

En Cuba es muy importante el factor humano, por lo que para cualquier institución es de suma importancia la comodidad de los trabajadores, puesto que conlleva a un mayor rendimiento en las tareas que ellos realizan. La implementación del sistema informático mejorará el control de la información referente a los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.

El sistema informático ayudara en gran medida la realización de las actividades laborales aumentando confiabilidad y rapidez en la gestión de la información y además el personal interactuará de forma dinámica con el sistema informático y las tecnologías en la actualidad.

Por lo antes dicho se concluye que es sostenible el “Sistema para la gestión de información de Medicina Natural y Tradicional en Granma”, de acuerdo a la dimensión socio-humanista.

Dimensión Ambiental

La interfaz que el sistema brinda al usuario resulta amigable, sin derroche de colores u otras molestias, interrelacionando así el trabajo con el entorno del lugar donde se utiliza el mismo. Se tuvieron en cuenta las exigencias fisiológicas del ser humano, es decir, tamaño de letra, espaciado entre caracteres.

Se recomienda, para evitar daños severos en la salud de los usuarios, que posean las condiciones de trabajo requeridas para el uso de productos informáticos: correcta iluminación, protectores de pantalla, asientos y escritorios cómodos y con las medidas correctas, ubicación adecuada del ordenador. Estas consideraciones evitarán molestias en la vista y en la postura de los implicados, efectos provocados por el tiempo que deben estar intercambiando información con la computadora.

Con la implantación del sistema informático no se contribuye de modo alguno al deterioro gradual del medio ambiente sino todo lo contrario, implantándolo se da un paso más en el cuidado del mismo por lo que se considera que es sostenible teniendo en cuenta la dimensión medio ambiental.

Dimensión Tecnológica

Es necesaria la capacitación del personal que interactuará con el software, esto se puede realizar a través del uso de un manual de usuario. El Centro de Medicina Natural y Tradicional cuenta con los medios necesarios para la correcta implantación y ejecución del sistema. Existe una estrecha comunicación entre el desarrollador de la aplicación y los usuarios finales, logrando que la aplicación que se desarrolle satisfaga las necesidades y expectativas de los clientes finales.

Para mantener el producto, los usuarios tendrán adjunto un respaldo de documentos y a su vez un Manual de usuario que describen como está confeccionada la aplicación Web. Para un mejor entendimiento los códigos están debidamente comentados.

Se recomienda realizar copias de seguridad tanto de la base de datos como de la aplicación en general ya que el sistema puede resultar vulnerable, desde el punto de vista de la ocurrencia de roturas o fallos del servidor.

Por todo lo antes expuesto se determinó que es sostenible desde el punto de vista tecnológico el desarrollo del “Sistema para la Gestión de Información de Medicina Natural y Tradicional en Granma”.

¿Es el sistema propuesto sostenible?

Estudiada la sostenibilidad según las dimensiones administrativas, socio-humanistas, ambientales y tecnológicas se puede afirmar que es sostenible el desarrollo del Sistema para la Gestión de Información de Medicina Natural y Tradicional en Granma la misma que permitirá dar respuesta en el menor tiempo posible, garantizando la calidad del sistema completamente.

2.4 Fase de Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.

El método escogido para realizar la estimación tiene como elemento el punto, el cual equivale a una semana perfecta de trabajo, esto se refiere a que solamente el equipo se dedica a labores relacionadas con la construcción del sistema sin la influencia o el retraso provocado por otros factores lo que en la práctica es complejo lograr (Guerrero, 1999).

2.4.1 Estimación de esfuerzos por Historias de Usuario

El cliente establece estimaciones del esfuerzo para implementar las historias de usuario, esto permite tener una medida real de la velocidad de progreso del proyecto y brindan una guía razonable a la cual ajustarse.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto

equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción.

Los resultados estimados se muestran seguidamente:

Tabla 2.5 Estimación de esfuerzos por Historia de usuario.

Historias de Usuario	Puntos estimados
Gestionar Cuenta de Usuario	0.3
Gestionar Trabajador	0.3
Gestionar Entidad	0.3
Gestionar Docente	0.3
Gestionar Diplomado	0.3
Gestionar Curso	0.3
Gestionar Centro Provincial	0.3
Gestionar Municipio	0.3
Gestionar Provincia	0.3
Gestionar Plan Producción	0.3
Gestionar Producto	0.3
Gestionar Tema	0.3
Gestionar Bibliografía Curso	0.3
Generar Reporte Extra Plan	0.1
Generar Reporte Matrícula	0.1
Generar Reporte Producción	0.1
Generar Reporte Diplomados	0.1

2.4.2 Plan de duración de las iteraciones

Una vez identificadas las historias de usuario del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas se procede a la planificación de la etapa de implementación del proyecto. Se decidió realizar la implementación de la aplicación en tres iteraciones, detalladas a continuación.

Tabla 2.6 Plan de duración de las iteraciones.

Iteraciones	Orden de la Historia de Usuarios	Duración de la Iteración
1ra Iteración	<ol style="list-style-type: none"> 1. Gestionar Cuenta de Usuario 2. Gestionar Trabajador 3. Gestionar Entidad 4. Gestionar Docente 5. Gestionar Diplomado 	2 Semanas
2da Iteración	<ol style="list-style-type: none"> 6. Gestionar Centro Provincial 7. Gestionar Curso 8. Gestionar Plan Producción 9. Gestionar Municipio 10. Gestionar Provincia 11. Gestionar Producto 12. Gestionar Tema 13. Gestionar Bibliografía Curso 	2 Semanas
3ra Iteración	<ol style="list-style-type: none"> 14. Generar Reporte Extra Plan 15. Generar Reporte Matrícula 16. Generar Reporte Producción 17. Generar Reporte Diplomados 	1 Semanas

2.4.3 Plan de entrega

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores). En la siguiente tabla se muestra el Plan de duración de entregas en el cual se especifican un aproximado de las fechas para cada iteración.

Tabla 2.7 Módulos e Historias de Usuario.

Módulos	Historia de Usuario
Usuario	1. Gestionar Cuenta de Usuario
	2. Gestionar Trabajador
	3. Gestionar Entidad
	4. Gestionar Docente
	5. Gestionar Centro Provincial
Negocio	6. Gestionar Diplomado
	7. Gestionar Curso
	8. Gestionar Plan Producción
	9. Gestionar Municipio
	10. Gestionar Provincia
	11. Gestionar Producto
	12. Gestionar Tema
	13. Gestionar Bibliografía Curso
Reportes	14. Generar Reporte Extra Plan
	15. Generar Reporte Matrícula
	16. Generar Reporte Producción
	17. Generar Reporte Diplomados

Tabla 2.8 Módulos e Iteraciones.

Módulos	1ra Iteración 2da semana Junio	2da Iteración 5ta semana Junio	3ra Iteración 2da semana Julio
Usuario	V1.0		Terminado
Negocio		V1.0	Terminado
Reportes		V1.0	Terminado

2.5 Iteraciones

Se considera esta fase la principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente debe participar activamente durante esta fase del ciclo.

1ra Iteración.

Esta iteración tendrá como objetivo darle cumplimiento a las historias de usuarios que tengan prioridad alta, estas historias de usuario son 1, 2, 3, 4 y 5 que son las de mayor importancia para el cliente, estas recogen las principales funcionalidades del sistema.

2da Iteración.

En esta iteración se le dará cumplimiento a las historias de usuario de prioridad media 6, 7, 8, 9, 10, 11, 12 y 13 que complementan las funcionalidades de las Historias de Usuario realizadas en la iteración anterior.

3ra Iteración.

La implementación de las historias de usuarios 14, 15, 16, 17 y 18 en esta iteración proporcionará una idea completa de la aplicación, al finalizar su

implementación quedará terminado el sistema.

2.5.1 Tareas

Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable. Estas historias de usuario son divididas en tareas de entre 2 y 6 días de duración que se asignarán a los programadores.

Las tareas restantes se muestran en el [Anexo 3](#).

Tabla 2.9 Tareas establecidas en cada iteración.

Iteraciones	Historias de Usuario	Tarea	Puntos Estimados
1ra Iteración	Gestionar Cuenta de Usuario	Diseño de interfaz para la selección de los usuarios	0.1
		Insertar usuarios	0.2
		Desactivar usuarios	
		Modificar usuarios	
	Gestionar Trabajador	Diseño de interfaz para la selección de Trabajador	0.1
		Buscar Trabajador	0.2
		Insertar Trabajador	
		Eliminar Trabajador	
		Modificar Trabajador	
	Gestionar Entidad	Diseño de interfaz para la selección de Entidad	0.1
		Buscar Entidad	0.2
		Insertar Entidad	
		Eliminar Entidad	
		Modificar Entidad	
	Gestionar Docente	Diseño de interfaz para la selección de Docente	0.1
		Buscar Docente	0.2
Insertar Docente			
Eliminar Docente			

Capítulo 2. Descripción, valoración y construcción de la propuesta de solución

		Modificar Docente	
	Gestionar Diplomado	Diseño de interfaz para la selección de Diplomado	0.1
		Buscar Diplomado	
		Insertar Diplomado	0.2
		Eliminar Diplomado	
		Modificar Diplomado	
2da Iteración	Gestionar Curso	Diseño de interfaz para la selección de Curso	0.1
		Buscar Curso	
		Insertar Curso	0.2
		Eliminar Curso	
		Modificar Curso	
	Gestionar Centro Provincial	Diseño de interfaz para la selección de Centro Provincial	0.1
		Buscar Centro Provincial	
		Insertar Centro Provincial	0.2
		Eliminar Centro Provincial	
		Modificar Centro	
	Gestionar Municipio	Diseño de interfaz para la selección de Municipio	0.1
		Buscar Municipio	
		Insertar Municipio	0.2
		Eliminar Municipio	
		Modificar Municipio	

Capítulo 2. Descripción, valoración y construcción de la propuesta de solución

	Gestionar Provincia	Diseño de interfaz para la selección de Provincia	0.1
		Buscar Provincia	0.2
		Insertar Provincia	
		Eliminar Provincia	
		Modificar Provincia	
	Gestionar Plan Producción	Diseño de interfaz para la selección de Plan Producción	0.1
		Buscar Plan Producción	0.2
		Insertar Plan Producción	
		Eliminar Plan Producción	
		Modificar Plan Producción	
	Gestionar Producto	Diseño de interfaz para la selección de Producto	0.1
		Buscar Producto	0.2
		Insertar Producto	
		Eliminar Producto	
		Modificar Producto	
Gestionar Tema	Diseño de interfaz para la selección de Tema	0.1	
	Buscar Tema	0.2	
	Insertar Tema		
	Eliminar Tema		
	Modificar Tema		
Gestionar Bibliografía Curso	Diseño de interfaz para la selección de Bibliografía Curso	0.1	

		Buscar Bibliografía Curso	
		Insertar Bibliografía Curso	0.2
		Eliminar Bibliografía Curso	
		Modificar Bibliografía Curso	
3ra Iteración	Generar Reporte Extra Plan	Mostrar Reporte Extra Plan	0.1
	Generar Reporte Matrícula	Mostrar Reporte Matrícula	0.1
	Generar Reporte Producción	Mostrar Reporte Producción	0.1
	Generar Reporte Diplomados	Mostrar Reporte Diplomados	0.1

Tareas de la Primera Iteración

Tabla 2.10 Tarea Diseño de la interfaz de Gestionar Cuenta de Usuario.

Tarea	
Número Tarea: 1	Número Historia: 1
Nombre Tarea: Diseño de la interfaz de Gestionar Cuenta de Usuario.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.1
Fecha de Inicio: 02/06/2013	Fecha Fin: 02/06/2013
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se diseñará una página para la gestión de los usuarios que pueden autenticarse en el sistema.	

Tabla 2.11 Tarea Insertar Usuario.

Tarea	
Número Tarea: 2	Número Historia: 1
Nombre Tarea: Insertar Usuario.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.067
Fecha de Inicio: 03/06/2013	Fecha Fin: 04/06/2013
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se implementará un método para que el administrador pueda insertar nuevos usuarios.	

Tabla 2.12 Tarea Desactivar Usuario.

Tarea	
Número Tarea: 3	Número Historia: 1
Nombre Tarea: Desactivar Usuario.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.067
Fecha de Inicio: 03/06/2013	Fecha Fin: 04/06/2013
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se implementará un método para que el administrador pueda desactivar el o los usuarios que considere.	

2.6 Producción

En la fase de producción se requiere de pruebas adicionales y revisiones de rendimiento antes de que las herramientas de enseñanza sean trasladadas al entorno de los usuarios. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

2.6.1 Diseño

Este sistema fue diseñado para facilitar la gestión de información de los procesos de producción y capacitación de forma dinámica y agradable al usuario. Para ello se emplearon algunas recomendaciones o premisas que XP establece a la hora de abordar esta etapa.

Es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace, tratar siempre de realizar las actividades de la manera más sencilla posible. Es así que el sistema informático desarrollado presenta un diseño simple y sencillo, orientado al entorno de trabajo de los usuarios para que se sienta identificado con la aplicación.

Este sistema fue diseñado para facilitar la gestión de información de los procesos de producción y capacitación de forma dinámica y agradable al usuario. Para ello se emplearon algunas recomendaciones o premisas que XP establece a la hora de abordar esta etapa.

Es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace, tratar siempre de realizar las actividades de la manera más sencilla posible. Es así que el sistema informático desarrollado presenta un diseño simple y sencillo, orientado al entorno de trabajo de los usuarios para que se sienta identificado con la aplicación.

Se utilizaron los colores verde y blanco ya que son los colores característicos de la institución. Se utilizó el color blanco para el fondo de las páginas, el negro para las letras garantizando una lectura favorable de los textos. Se usan las letras Arial, Helvetica, sans-serif para los textos de las páginas. Estos tipos de letra permiten una lectura rápida y cómoda

Para evitar largos tiempos de espera a la hora de cargar la página y visualizarla se utilizó el mínimo uso de imágenes y animaciones.

Para la construcción del sistema se tomaron en cuenta algunos de los estándares de implementación propuestos: un header, donde se muestra, el logo de la institución, el nombre del sistema e imágenes y textos que muestren de manera general el menú de opciones con sus respectivos vínculos del sistema informáticos, y la pantalla principal donde se visualiza la lista de usuarios registrados.

Algunas imágenes del sistema informático se pueden consultar en el [Anexo 1](#).

El diseño de la base de datos fue realizado con la herramienta MySQL, el mismo está compuesto por 22 tablas, la cuales están normalizadas,

cumpliendo con las normas establecidas para el diseño de bases de datos. En el [Anexo 4](#) puede ser visto el modelo de datos de la aplicación.

Tarjetas CRC

La metodología XP para el diseño de las aplicaciones no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

La forma de diseño y organización que se adopta es de diseñar una tarjeta CRC (Clase-Responsabilidad-Colaboración) por cada uno de los módulos que brindan una funcionalidad directa al negocio, es decir aquellos que fueron desarrollados desde la raíz. De esta forma se obtiene un diseño simple y no se implementan características que no son necesarias. Estas tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos.

Tabla 2.13 Tarjeta CRC Módulo Curso

Módulo Curso	
Funcionalidades	Colaboraciones
Insertar Curso	Módulo Trabajador Módulo Docente
Modificar Curso	
Eliminar Curso	
Mostrar Curso	
Buscar Curso	
Exportar Curso	

Tabla 2.14 Tarjeta CRC Módulo Tema

Módulo Tema	
Funcionalidades	Colaboraciones
Insertar Tema	Módulo Curso
Modificar Tema	
Eliminar Tema	
Mostrar Tema	
Buscar Tema	
Exportar Tema	

Las tarjetas CRC restantes se pueden consultar en el [Anexo 5](#).

2.6.2 Codificación

Para la implementación del sistema informático se utiliza el lenguaje de programación web por el lado del servidor PHP (*Personal Home Page*), con su *framework* CodeIgniter y es diseñado originalmente para la creación de aplicaciones web dinámicas que emplean clases en el código fuente porque según las características del sistema informático se considera que es necesario utilizar la programación orientada a objetos.

2.6.3 Prueba

En la metodología XP uno de los aspectos más importantes es el proceso de pruebas, en el cual los desarrolladores le realizarán estas constantemente al producto, esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. La metodología XP divide las pruebas del sistema en dos grupos:

Pruebas unitarias

Las pruebas unitarias aseguran que un único componente de la aplicación produce una salida correcta para una determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada

caso particular. Las pruebas unitarias se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto.

Pruebas de aceptación

Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación.

Las pruebas de aceptación son pruebas de caja negra que se realizan a partir de las historias de usuarios (Guerrero, 1999). Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas.

A continuación se muestran algunas de las pruebas de aceptación propuestas a realizarse, para las pruebas restantes consultar [Anexo 6](#).

Tabla 2.15 *Caso de prueba Diseño correcto de la interfaz para la selección de los usuarios.*

Caso de Prueba	
Código de Prueba: 1	Número de Historia de Usuario: 1
Nombre de la Prueba: Diseño correcto de la interfaz para la selección de los usuarios.	
Descripción: Prueba para verificar el diseño correcto de la interfaz para la selección de los usuario del sistema. El administrador es el encargado de de El Especialista económico accede a la opción de área, modificar área, entrar los nuevos datos correctos y válidos, el sistema notifica: El área se modificó correctamente.	

Condiciones de ejecución: El área debe estar en la base de datos
Entrada: Datos requeridos para la modificación del área válidos.
Resultado esperado: El área se modificó correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 2.16 Caso de prueba Insertar correctamente usuario

Caso de Prueba	
Código de Prueba: 2	Número de Historia de Usuario: 1
Nombre de la Prueba: Insertar correctamente usuario.	
Descripción: Prueba para verificar el ingreso correcto de un nuevo usuario dentro del sistema. El administrador es el encargado de acceder correctamente un usuario y asignarle los privilegios que tendrá dentro del sistema	
Condiciones de ejecución: El usuario no debe estar en la base de datos	
Entrada: Datos requeridos para el ingreso de usuario válidos.	
Resultado esperado: El usuario se insertó correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 2.17 Caso de prueba error insertar usuarios.

Caso de Prueba	
Código de Prueba: 3	Número de Historia de Usuario: 1
Nombre de la Prueba: error insertar usuarios.	
Descripción: Prueba para verificar el error al ingreso de un nuevo usuario dentro del sistema. El administrador accede a la opción de crear un nuevo usuario donde llenara los campos requeridos por el sistema y se mostrara un mensaje del error en el sistema.	
Condiciones de ejecución: El usuario no debe estar en la base de datos.	

Entrada: Datos requeridos para el ingreso del usuario válidos.

Resultado esperado: El usuario no se puede insertar en el sistema, este campo solo acepta letras.

Evaluación de la prueba: Prueba satisfactoria.

2.7 Mantenimiento

La fase de mantenimiento puede requerir cambios en la estructura de confección de la aplicación. El proyecto XP debe mantener la aplicación en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones y para que esto se pueda realizar requiere de tareas de soporte para el cliente.

Mientras se desarrolló el sistema informático se tuvo en cuenta las opiniones del cliente para el cual fue implementado, de esta forma en esta etapa fueron identificadas cada una de las necesidades novedosas del usuario que pudieron propiciar cambios en el sistema informático.

2.8 Muerte del Proyecto

Se da cuando el cliente no tiene más historias para ser incluidas en la aplicación. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad de la aplicación y no se realizan más cambios en la arquitectura de la misma.

2.9 Conclusiones del capítulo

El análisis del funcionamiento del sistema de control de información de los procesos de producción y capacitación de Medicina Natura y Tradicional en Granma permitió definir las características necesarias para la creación de la aplicación web acorde a las necesidades del cliente, todo esto a través de la metodología de desarrollo software escogida. Es así como en este capítulo se especifican los resultados de cada una de las fases que XP propone. Se construyó la aplicación Web que introdujo una nueva vía para gestionar la información de información de los procesos de producción y capacitación de Medicina Natura y Tradicional, lo cual favoreció a solucionar los problemas que originaron esta investigación.

Conclusiones Generales

El desarrollo del Sistema para la Gestión de información de Medicina Natural y Tradicional en Granma da cumplimiento a los objetivos de la investigación, por lo que se puede concluir que:

- Se caracterizó el proceso de gestión de la información de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma, lo que facilitó la búsqueda de la solución a la problemática planteada.
- Se implementó un sistema informático para la gestión de la información de los procesos de producción y capacitación del Centro de Medicina Natural y Tradicional en la provincia Granma.

Recomendaciones

- ✓ Generalizar el sistema informático resultado de la presente investigación a otras instituciones similares del país.
- ✓ Presentar los resultados de la investigación en eventos científicos.
- ✓ Consultar la presente investigación como material de estudio para la realización de otros similares.

Referencias bibliográficas

- Aguilar, M. (2013). Gestión de la Información y el conocimiento. Recuperado, de <http://www.archiveros.info/group/archivexperto/forum/topics/la-gesti-n-de-la-informaci-n-y-el-conocimiento>
- Aldana, C. (2009). *Apache El servidor Web más reconocido*.
- Alvarez, M. (2001). Introducción a los lenguajes web Desarrolladores Web. Recuperado, de <http://www.desarrolloweb.com/contacta/1.html>
- Angel, G. (2010). Sistema Gestor de Base de Datos. Correspondiente al 13/04/2013.
- Arias, M. (2008). Definición de lenguaje de programación Available from <http://catedraprogramacion.foroactivos.net/t83-definicion-de-lenguaje-de-programacion>
- Askallorg, S. (2007). Programando en Java.
- Bertino E y Martino A (1995). *Sistemas de bases de datos orientadas a objetos* (Díaz de Santos ed. Vol. 2).
- Betancourt, R. (2010). *DEFINICIÓN Y TIPOS DE APLICACIONES WEB*. 27 Correspondiente al 27/05/2010.
- Brusilovsky P, Gilles G, Frasson C y VanLehn K (2007). *Proceedings of Intelligent Tutoring Systems*. (Springer Verlag ed.).
- Cabrera, A. (2010). MySQL vs PostgreSQL. Recuperado, de <http://www.bisente.com/documentos/mysql-postgres.html>
- Canós J, Letelier P y Penadés C (2003). *Metodologías Ágiles en el Desarrollo de Software*.
- Carrasco, P. (2009). CodeIgniter un poderoso framework open source - Desarrollo Zona Linux. Recuperado, de <http://www.codeigniter.com>
- Cinca, T. (2008). *Características principales de la Web. Web Estática y Web Dinámica*.
- Cobo Á, Gómez P, Pérez D y Rocha R (2010). *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. Ciencias de la información (Vol. 13).
- Cobo, Á. y Gómez, P. (2005). *PHP y MySQL- tecnologías para el desarrollo de aplicaciones web* (Díaz de Santos ed. Vol. 2).
- Colombo B y Demetrescu C (2003). *A Java-based System for Building Animated Presentations over the Web* (Vol. 1).

- Corona, G. (2010). *Sistema para la gestión del cuestionario Kano en la ECTA "26 de Julio de la Provincia Granma*. Universidad de Granma, Bayamo.
- Crespo P y Garcia P (2000). Frameworks. Recuperado, de <http://es.wikipedia.org/wiki/Framework>
- De la Torre, A. (2006). *Servidores Web* (Vol. 2).
- Díaz, J. (2009). Las metodologías ágiles como garantía de calidad del software. 5, 40-43,
- Eloi, P. (2009). Tipos de datos en PHP. Correspondiente al 15/09/2011.
- Furgeta, A. (2002). *Introduction to Systems Engineering* (MIT Press. ed.). New York.
- García, J. (2006). Manual de PHP. 57.
- Guerra J y Guerra D (2011). *Sistema de Gestión para el Control de los Procesos de Produccion y capacitación de la Medicina Natural y Tradicional en Granma*. Tesis de Trabajo de diplomado, Universidad de Granma, Bayamo.
- Guerrero, J. (1999). Metodologías Ágiles de desarrollo de software XP Fases.
- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software* (Addison-Wesley Company ed.). Stanford.
- JBC (2009). Manual de JBC (Java Byte Code).
- Joskowicz, J. (2008). *Reglas y Prácticas en eXtreme Programming*: 10-02-2008.
- Labs, S. (2012). Symfony. Recuperado el 20/04/2013, de <http://www.symfony.com>
- Leonard, E. (2011). *Sistema para la Gestión de Información Relacionada con el grupo de Operación y Mantenimiento Técnico del Centro Telefónico de ETECSA Holguín*. Universidad de Granma, Bayamo.
- Lockhart, T. (1996). Tutorial de PostgreSQL.,
- Málaga, D. (2008). *Manual de HTML y JavaScript*.
- MasAdmin (2009). Software para la Micro, Pequeña y Mediana Empresa. Recuperado, de <http://www.masadmin.com>
- Oliver, K. (2000). *Methods for developing constructivist learning on the web*. *Educational Technology Research & Development* (Vol. 6).
- Petckvic, D. (2005). Microsoft SQL Server 2005 McGraw-Hill Professional. Recuperado

- Prentzas J, Hatzilygeroudis I y Garofalakis J (2002). *A web Based Intelligent Tutoring System Using Hybrid Rules* (Vol. 1).
- Pressman, R. (2002). *Ingeniería del Software* (Quinta ed.). Madrid.
- Ruiz, H. (2006). *Programación web avanzada* (Félix Varela ed.). La Habana.
- Vásquez, I. (2008). Clientes delgados, el advenimiento de la computación en red. Recuperado, de http://www.mordecki.com/Otros_Temas/clientesdelgados/clientesdelgados.shtml
- Wed, G. (2013). Tipos de Lenguajes de Programación. Correspondiente al 02/03/2013.
- Wilsoft, N. (2011). Software para Evaluación de Desempeño y Control de Capacitación. Recuperado, de <http://www.wilsoft-la.com>
- Zelkovitz M, Shaw A y Gannon J (2009). *Principles of Software Engineering and Design* (MIT Press ed. Vol. 2).
- Zuarez, T. (2009). Programmer's Reference Guide, Zend Framework. Recuperado, de <http://framework.zend.com/manual/en/index.html>

Bibliografía

A. Navasa, M.A. Pérez, M. Sánchez. Ed. M. Sánchez. 1999. Aplicación de UML al desarrollo de sistemas orientados a objetos. 1999. 84-605-9632-X.

Alexander, Christopher, Ishikawa, Sara y Silverstein, Murray. 1977. A Pattern Language. New York: Oxford University Press, 1977. 0195019199.

Clements, Paul, y otros. 2002. Documenting Software Architectures: Views and Beyond. s.l.: Addison Wesley, 2002.0-201-70372-6.

Cockburn, Alistair. 2000. Writing Effective Use Cases. s.l.: Addison-Wesley, 2000. 0201702258.

Coplien, James. 1996. Software Patterns. s.l.: SIGS, 1996. 978-1884842504.

Delgado, Andrea, y otros. 2008. Metodologías de desarrollo para Service Oriented Architectures con Rational Unified Process. Monte Video, Uruguay: s.n., 2008. 4. Bass, Len, Clements, Paul y Kazman, Rick. 2003. Software Architecture in Practice, Second Edition. s.l.: Pearson Education, Inc., 2003. ISBN: 0-321-154959.

Billy Reinoso, Carlos y Kicillof, Nicolás. 2004. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Buenos Aires: s.n., 2004

Grady Booch, Robert C, Martin y James Newkirk. Oriented Analysis and Design with Applications. s.l. : Addison Wesley Longman, 1998.

José H. Canós, Patricio Letelier y M^a Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software.

Stephens, Matt y Rosenberg, Doug. Extreme Programming Refactored: The Case Against XP. Apress, 2003.

Jorge Ferrer Zarzuela. Metodologías Ágiles.

Joskowicz. Reglas y Prácticas en eXtreme Programming. 2008.

PostgreSQL. [En Línea] <http://www.postgresql.org>

PostgreSQL. Características fundamentales [En Línea].

http://www.netpecos.org/docs/mysql_postgres/x15.html

PostgreSQL Comparativas. [En Línea].

http://www.netpecos.org/docs/mysql_postgres/x108.html

Miguel, Angel Alvarez. desarrolweb.com. [En Línea] [Revisado: Marzo 6, 2011.] <http://www.desarrolloweb.com/articulos/que-es-html.html>

maestrosdelweb.com. [En Línea] [Revisado: Marzo 6, 2011.]

<http://www.maestrosdelweb.com/editorial/introcss/>.

Javier, García Gallego Felipe. css.infames.org. [En Línea] [Revisado: Marzo 7, 2011.] <http://css.infames.org/ventajas.html>.

Anexos

Anexo 1 Imágenes de la solución propuesta

Imagen de acceso al Sistema

Login

Por favor entre a continuación su correo y contraseña.

Correo:

Contraseña:

Recordar Contraseña:

Login

Imagen de la gestión de Cursos

Sistema de Gestión de Información del Centro de Medicina Natural y Tradicional

Inicio Configuración Capacitación Administración Producción Reportes Salir

Región	Conocimientos	Habilidades	Evaluación	Medios de enseñanza	Créditos	Semanas de duración	Fecha de inicio	Horas teóricas	Horas prácticas	Horas de evaluaciones	Cantidad plazas	Requerimiento de ingreso	Alumnos	Bibliografía	Docentes	Acciones
Región	Conocimiento	Habilidad	Escrita	Computadora	3	4	10/10/2013	10	20	4	20	Conocer	Yoel Bentez Martinez	Mi libro	Eric Ismael Leonard Brizuela	
Región	Conocimiento	Habilidad	Escrita	Computadora	2	4	12/05/2013	2	1	2	10	Título universitario	Yoel Bentez Martinez	Mi libro	Eric Ismael Leonard Brizuela	

Mostrando 1 a 2 de 2 registros

©Copyrights 2013 Myriam Janeth Lamingo Chingo Ana Lucia Alvarez Jacho Universidad de Granma Facultad de Ciencias Técnicas

Imagen para la gestión de Plan Producción

Sistema de Gestión de Información del Centro de Medicina Natural y Tradicional

Inicio Configuración Capacitación Administración Producción Reportes Salir

Agregar Plan de producción Exportar Imprimir

Fecha	Entidad	Producto	Plan	Real	Cumplimiento	Causas de incumplimiento	Acciones
13/06/2013	Universidad de Granma	Alcohol etílico	35.00	35.00	100%		
10/07/2013	Universidad de Granma	Alcohol etílico	123.00	120.00	97.56%	Falta de electricidad	
11/07/2013	Líquidos orales	Alcohol etílico	43.00	40.00	93.02%		
11/07/2013	Líquidos orales	Alcohol etílico	20.00	10.00	50%		
11/07/2013	Líquidos orales	Alcohol etílico	10.00	10.00	100%		
03/07/2013	Universidad de Granma	Alcohol etílico	140.00	140.00	100%		
11/07/2013	Universidad de Granma	Crema natural	24.00	7.00	29.17%	Falta de agua	

Buscar: [] Buscar todos [] Buscar [] Limpiar filtros []

25 [] [] [] [] Página 1 de 1 [] [] registros

©Copyrights 2013 Myriam Janeth Lamingo Chingo Ana Lucia Álvarez Jacho Facultad de Ciencias Técnicas

Imagen del reporte Producción Producto

Sistema de Gestión de Información del Centro de Medicina Natural y Tradicional

Inicio Configuración Capacitación Administración Producción Reportes Salir

Exportar Imprimir

Año	Mes	Producto	Plan mensual	Real	% de cumplimiento
2013	6	Alcohol etílico	35.00	35.00	100.00
2013	7	Alcohol etílico	336.00	320.00	97.56
2013	7	Crema natural	24.00	7.00	29.17

25 [] [] [] [] Página 1 de 1 [] [] Mostrando 1 a 3 de 3 registros

©Copyrights 2013 Myriam Janeth Lamingo Chingo Ana Lucia Álvarez Jacho Facultad de Ciencias Técnicas

Anexo 2 Historias de usuarios

Tabla 2.1 Historia de Usuario Generar Docente

Historia de Usuario	
Número: 4	Usuario: Administrador, Capacitador.
Nombre: Gestionar Docente	
Prioridad en Negocio: Alta	Puntos Estimados: 0.3
Riesgos en Desarrollo: Alta	Iteración Asignada: 1
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se podrá insertar, modificar, eliminar y buscar los datos de los docentes.	
Observaciones:	

Tabla 2.2 Historia de Usuario Gestionar Diploma

Historia de Usuario	
Número: 5	Usuario: Administrador, Capacitador.
Nombre: Gestionar Diplomado	
Prioridad en Negocio: Alta	Puntos Estimados: 0.3
Riesgos en Desarrollo: Alta	Iteración Asignada: 2
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Podrá insertar, modificar, eliminar y buscar diplomados.	
Observaciones:	

Tabla 2.3 Historia de Usuario Gestionar Curso

Historia de Usuario	
Número: 6	Usuario: Administrador, Capacitador.
Nombre: Gestionar Curso	
Prioridad en Negocio: Media	Puntos Estimados: 0.3
Riesgos en Desarrollo: Media	Iteración Asignada: 2
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se podrá insertar, modificar, eliminar y buscar nuevos cursos.	
Observaciones:	

Anexos 3 Tareas para realización de las Historias de Usuario

Tabla 3.1 Tarea # 4 Tarea Modificar Usuarios

Tarea	
Número Tarea: 4	Número Historia: 1
Nombre Tarea: Modificar Usuarios	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.067
Fecha de Inicio: 03/06/2013	Fecha Fin: 04/06/2013
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se implementará un método para que el administrador pueda modificar los datos de los usuarios del sistema.	

Tabla 3.2 Tarea # 1 Diseño de interfaz para la selección de Trabajador

Tarea	
Número Tarea: 1	Número Historia: 2
Nombre Tarea: Diseño de interfaz para la selección de Trabajador	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.067
Fecha de Inicio: 05/06/2013	Fecha Fin: 05/06/2013
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se diseñará una página para la gestión de los trabajadores.	

Tabla 3.3 Tarea # 2 Buscar Trabajador

Tarea	
Número Tarea: 2	Número Historia: 2
Nombre Tarea: Buscar Trabajador.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.067
Fecha de Inicio: 05/06/2013	Fecha Fin: 05/06/2013
Programador Responsable: Myriam Janeth Lamingo Chingo, Ana Lucia Álvarez Jacho.	
Descripción: Se implementará un método para que el especialista de capacitación pueda buscar los datos de los usuarios que se encuentran registrados en el sistema	

Anexo 5 Tarjetas CRC

Tabla 5.1 Tarjeta CRC Módulo Docente

Módulo Docente	
Funcionalidades	Colaboraciones
Insertar Docente	
Modificar Docente	
Eliminar Docente	
Mostrar Docente	
Buscar Docente	
Exportar Docente	

Tabla 5.2 Tarjeta CRC Módulo Bibliografía

Módulo Bibliografía	
Funcionalidades	Colaboraciones
Insertar Bibliografía	
Modificar Bibliografía	
Eliminar Bibliografía	
Mostrar Bibliografía	
Buscar Bibliografía	
Exportar Bibliografía	

Tabla 5.3 Tarjeta CRC Módulo Diplomado

Módulo Diplomado	
Funcionalidades	Colaboraciones
Insertar Diplomado	Módulo Trabajador
Modificar Diplomado	
Eliminar Diplomado	
Mostrar Diplomado	
Buscar Diplomado	
Exportar Diplomado	

Tabla 5.4 Tarjeta CRC Módulo Trabajador

Módulo Trabajador	
Funcionalidades	Colaboraciones
Insertar Trabajador	Módulo Curso Módulo Diplomado Módulo Municipio
Modificar Trabajador	
Eliminar Trabajador	
Mostrar Trabajador	
Buscar Trabajador	
Exportar Trabajador	

Tabla 5.5 Tarjeta CRC Módulo Municipio

Módulo Municipio	
Funcionalidades	Colaboraciones
Insertar Municipio	Módulo Provincia
Modificar Municipio	
Eliminar Municipio	
Mostrar Municipio	
Buscar Municipio	
Exportar Municipio	

Tabla 5.6 Tarjeta CRC Módulo Provincia

Módulo Provincia	
Funcionalidades	Colaboraciones
Insertar Provincia	
Modificar Provincia	
Eliminar Provincia	
Mostrar Provincia	
Buscar Provincia	
Exportar Provincia	

Anexo 6 Pruebas de aceptación realizadas al sistema

Tabla 6.1 Caso de Prueba Error insertar usuarios

Caso de Prueba	
Código de Prueba: 4	Número de Historia de Usuario: 1
Nombre de la Prueba: error insertar usuarios.	
Descripción: Prueba para verificar el error al ingreso de un nuevo usuario dentro del sistema. El administrador accede a la opción de crear un nuevo usuario donde llenara los campos requeridos por el sistema y se mostrara un mensaje del error en el sistema.	
Condiciones de ejecución: El usuario no debe estar en la base de datos.	
Entrada: Datos requeridos para el ingreso del usuario válidos.	
Resultado esperado: El usuario no se puede insertar en el sistema, este campo solo acepta letras.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.2 Caso de Prueba Desactivación usuario

Caso de Prueba	
Código de Prueba: 5	Número de Historia de Usuario: 1
Nombre de la Prueba: Desactivación correcta usuario.	
Descripción: Prueba para verificar la Desactivación correcta usuario. Dentro del sistema. El administrador acceder a la opción desactivar usuario y selecciona desactivar usuario notificando acción realizada con éxito	
Condiciones de ejecución: El usuario debe estar en la base de datos	
Entrada: Datos requeridos para desactivar de usuario válidos.	
Resultado esperado: El usuario se desactivo correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 6.3 Caso de Prueba Desactivación incorrecta usuario

Caso de Prueba	
Código de Prueba: 6	Número de Historia de Usuario: 1
Nombre de la Prueba: Error al desactivar usuarios.	
Descripción: Prueba para verificar la funcionalidad de desactivar usuarios del sistema. El administrador accede a la opción desactivar y selecciona los datos requeridos; el sistema detecta un error de conexión y notifica: "Acción fallida - Error de conexión".	
Condiciones de ejecución: El usuario debe existir en la base de datos.	
Entrada: Datos requeridos para desactivar usuario válido.	
Resultado esperado: desactivar usuario no incorrecto.	
Evaluación de la prueba: Prueba satisfactoria.	