

Universidad de las Ciencias Informáticas

Facultad 4



Sistema para centralizar la información de Recursos Humanos en la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación.

Trabajo para optar por el Título de Ingeniero en Ciencias Informáticas

Autor: Rajiv Rondón Padilla

Tutores: Ing. Héctor Luis Reyes Pupo

Lic. Ulises Llorente Pérez

Declaración de autoría

Declaro que soy autor único de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga uso pertinente de este trabajo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del 2012.

Autor:

Rajiv Rondón Padilla

Tutores:

Ing. Héctor Luis Reyes Pupo.

Lic. Ulises Llorente Pérez.

Pensamientos

El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres.

Steve Jobs

Milito en el bando de los impacientes, milito en el bando de los apurados, de los que siempre presionan para que las cosas se hagan y de los que muchas veces tratan de hacer más de lo que se puede.

Fidel Castro

Dedicatorias

A mi madre por ser la mejor de todas, ejemplo de abnegación y constancia, por ser lo único que tengo en la vida y la única razón de mí ser. Fuente de amor inagotable que me incita cada día a seguir adelante, guerrera en tiempos difíciles y luz que me ilumina.

A la memoria de mi padre que la vida no le permitió verme graduado.

A las tías que más quiero, Juanita, Julia y Magalys por darme fuerzas y ser tan buenas conmigo.

A mi familia de crianza, que aunque no tengamos la misma sangre han estado presente en mi vida en todo momento y nunca me han dado la espalda.

A mi segunda madre Maty y a mi segundo padre Rafael.

A mis hermanas Josefa y Juana, a mi preciosa Gabriela, a Gladys y Leonardo por ser tan buenos conmigo, solo espero algún día hacer por todos ustedes lo que han hecho por mí, no tengo como agradecerles.

Agradecimientos

A mis tutores Héctor y Ulises por ayudarme tanto.

A mi antiguo grupo 8109 por darme la posibilidad de conocer lo que es estar becado y mostrarme cosas que realmente cambiaron mi vida, especialmente a Yasmany Sánchez, Yuniel Perdomo y Julio Castro, por ser mis primeros amigos en esta escuela.

A dos personitas que quise y quiero mucho, Lisandra Pompa y Lianet Cabo por ser tan especiales en mi vida, las recordaré por siempre.

A mis nuevos amigos de 5to año, René, Arelis, Héctor, Maday, Carlos, Miriam, Mayelín, Remberto, Mayara y Ramiro por estar presentes en cada momento del desarrollo de esta tesis.

A los amigos de siempre (UCI-Camagüey): Carlos Morciego, Dayán Trujillo y Daríel Carmenates. Si se me queda alguno perdónenme.

A mi ex novia Ángela Echevarría por cambiar mi vida y por hacerme feliz en tan poco tiempo, por enseñarme lo que es el amor, porque realmente no lo sabía, lástima que estés tan lejos y sin embargo tan cerca de mi corazón. Espero seas una gran doctora.

A alguien que no diré el nombre, ella sabe quién es, gracias por soportarme y pedirme que nos casáramos, realmente no estaba preparado para eso, discúlpame, no fui la mejor persona contigo, pero te quise mucho y me enamoré de ti aunque me haya dado cuenta muy tarde, gracias por apoyarme siempre y por ser la primera persona que creyó en mí. Sabes que te amaré siempre.

Resumen

El presente trabajo contiene la investigación, desarrollo y solución de una aplicación web dinámica con el objetivo de optimizar el trabajo referente al proceso de gestión de información de Recursos Humanos en la Dirección de Producción Audiovisual (DPA) del Centro de Tecnologías para la Formación (FORTES). Para ello fue necesario realizar un análisis del negocio, aspecto que contribuyó a la creación de la propuesta. El modelado del sistema fue elaborado siguiendo los principios y recomendaciones de la metodología de desarrollo de software Programación Extrema (XP), la cual constituyó una entrada fundamental para la implementación, de esta manera se garantizó la construcción de una aplicación que satisface las necesidades del cliente.

El trabajo incluye un estudio del estado del arte, las herramientas utilizadas, características del sistema, planificación, diseño, implementación y la estrategia de pruebas que se llevó a cabo. Además, un conjunto de conclusiones y recomendaciones que pueden ser de gran importancia para el tema tratado.

Palabras claves: Recursos Humanos, DPA, FORTES.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Conceptos asociados al problema de investigación	5
1.2.1 Sistemas de gestión de información.....	5
1.2.2 Gestión de Recursos Humanos.....	5
1.3 Estudio de sistemas similares.	5
1.3.1 Sistemas de Gestión de Recursos Humanos a nivel internacional.	6
1.3.2 Sistema de Gestión de Recursos Humanos a nivel nacional.....	7
1.3.3 Sistemas de Gestión de Recursos Humanos en la UCI.....	8
1.4 Herramientas y tecnologías informáticas.....	9
1.4.1 Metodologías para el desarrollo de software.	9
1.4.2 Lenguaje de modelado	13
1.4.3 Herramientas CASE (Ingeniería de software asistida por computadora)	14
1.4.4 Lenguajes de Programación.....	15
1.4.5 IDEs (Entorno de desarrollo integrado).....	18
1.4.6 Frameworks (Marco de trabajo).....	20
1.4.7 Servidores Web.....	23
1.4.8 Gestores de Base de Datos.....	24
1.5 Conclusiones del capítulo	26
Capítulo 2: Características del Sistema	27
2.1 Introducción:	27
2.2 Objeto de automatización.....	27
2.3 Información que se maneja.	28
2.4 Propuesta de sistema.....	28
2.5 Personas relacionadas con el sistema.	29
2.6 Exploración	29

2.6.1 Lista de Reserva del Producto (LRP).	29
2.7 Planificación.....	35
2.7.1 Plan de iteraciones.....	36
2.7.2 Plan de duración de las iteraciones.....	37
2.7.3 Plan de entregas	38
2.8 Diseño del Sistema	38
2.8.1 Arquitectura del sistema	38
2.8.2 Patrones de Diseño.....	39
2.8.3 Tarjetas CRC (Contenido – Responsabilidad - Colaboración).....	42
2.8.4 Diseño de la Base de Datos.	44
2.9 Conclusiones del capítulo	44
Capítulo 3: Implementación y Pruebas	45
3.1 Introducción	45
3.2 Estándar de codificación	45
3.2.1 Importancia de la utilización de estilos de código	45
3.2.2 Elementos generales que conforman un estilo de codificación.....	45
3.3 Fase de implementación	46
3.3.1 Iteración 1	47
3.3.2 Iteración 2	49
3.3.3 Iteración 3	52
3.4 Diagrama de Despliegue.....	56
3.5 Pruebas.....	56
3.5.1 Pruebas de aceptación.....	57
3.6 Conclusiones del capítulo	57
Conclusiones Generales:.....	58
Recomendaciones:.....	59
Referencias Bibliográficas.....	60
Bibliografía	63

Glosario de Términos:..... 67

Índice de Figuras

Figura 3 Diagrama de Despliegue.....	56
--------------------------------------	----

Índice de Tablas

Tabla 1: Personas relacionadas con el sistema.....	29
Tabla 2: Lista de Reserva del Producto (LRP).....	30
Tabla 3 HU Autenticar usuario.....	34
Tabla 4 HU Gestionar cuentas.	34
Tabla 5 HU Gestionar trabajadores.....	34
Tabla 6 HU Realizar búsquedas filtradas.	35
Tabla 18 Estimación de esfuerzos por historias de usuario.....	35
Tabla 19 Plan de duración de las iteraciones.	37
Tabla 20 Plan de entregas.	38
Tabla 21 Tarjeta CRC.....	42
Tabla 22 Tarjeta CRC Módulo Autenticar.....	43
Tabla 23 Tarjeta CRC Módulo Administrador.	43
Tabla 24 Tarjeta CRC Módulo Usuario.....	43
Tabla 25 Historias de Usuario abordadas en la primera iteración.....	47
Tabla 26 Tarea 1 HU Autenticar.....	47
Tabla 27 Tarea 2 HU Autenticar.....	48
Tabla 28 Tarea 3 HU Gestionar cuentas.	48
Tabla 29 Tarea 4 HU Gestionar trabajador	48
Tabla 30 Tarea 5 HU Gestionar trabajador	49
Tabla 31 Tarea 6 HU Realizar búsquedas filtradas.....	49
Tabla 32 Historias de Usuario abordadas en la segunda iteración.....	49
Tabla 33 Tarea 7 HU Controlar asistencia diariamente.	50
Tabla 34 Tarea 8 HU Controlar asistencia diariamente.	50
Tabla 35 Tarea 9 HU Gestionar salario.	50

Tabla 36 Tarea 10 HU Gestionar salario.....	51
Tabla 37 Tarea 11 HU Reportar evaluación de desempeño.....	51
Tabla 38 Tarea 12 HU Exportar modelos.....	51
Tabla 39 Historias de Usuario abordadas en la segunda iteración.....	52
Tabla 40 Tarea 13 HU Exportar actas.....	52
Tabla 41 Tarea 14 HU Gestionar capacitación.....	52
Tabla 42 Tarea 15 HU Gestionar capacitación.....	53
Tabla 43 Tarea 16 HU Gestionar estudios.....	53
Tabla 44 Tarea 17 HU Gestionar estudios.....	53
Tabla 45 Tarea 18 HU gestionar sanciones.....	54
Tabla 46 Tarea 19 HU gestionar sanciones.....	54
Tabla 47 Tarea 20 HU Gestionar estimulación moral.....	54
Tabla 48 Tarea 21 HU Gestionar estimulación moral.....	55
Tabla 49 Tarea 22 HU Envío de correo electrónico a los trabajadores.....	55
Tabla 50 Tarea 23 HU Publicación de trabajos, informes y planillas.....	55

Introducción

En la era de la información, las organizaciones requieren agilidad, movilidad, innovación y cambios necesarios para enfrentar las nuevas amenazas y oportunidades en un ambiente de intensa transformación y turbulencia. Los productos y servicios se adaptan de manera continua a las exigencias y necesidades de los clientes. Las personas, junto con sus conocimientos y habilidades intelectuales, se convierten en la base principal de la nueva organización.

La época actual aparece como un escenario que impone, como principal factor de competitividad, una gestión eficaz de los Recursos Humanos; lo que hace que una organización sea diferente de otra, condiciona su supervivencia y posibilita una mayor eficiencia en los diferentes procesos de la empresa. Gestionar dicha información se ha convertido en un proceso difícil, pues la capacidad de generación y almacenamiento de este recurso empresarial ha crecido significativamente a nivel mundial, lo que dificulta su análisis por los métodos tradicionales existentes, de ahí el surgimiento de nuevas técnicas y herramientas como los sistemas de gestión de información.

Los sistemas de gestión de información han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones, ejercen una influencia notable en la cultura organizacional, modelos laborales, funcionamiento y lo más importante, su implantación logra ventajas competitivas.

Internacionalmente las empresas que operan en el siglo XXI experimentan una aceleración en la evolución tecnológica y social que demanda la permanente necesidad de la asimilación de nuevas tecnologías, procesos de trabajo y procedimientos de gestión, enfrentándose a retos como la: rentabilidad, competitividad, globalización, velocidad de los cambios, capacidad de adaptación, crecimiento y tecnología.

El desarrollo de software en Cuba se ha visto incrementado en los últimos años, como parte del auge tomado también a nivel mundial, de ahí el surgimiento de centros de desarrollo de software a lo largo de todo el país.

En la Universidad de las Ciencias Informáticas, la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación, consta de una aplicación de escritorio que permite el control de los Recursos Humanos, pero dicho sistema está muy lejos de ser óptimo y no cumple con las expectativas de la entidad, ya que la versión existente no es multiplataforma y no tiene razón de ser con el proceso de migración a software libre que lleva la universidad para lograr la independencia tecnológica, la información no se encuentra centralizada y actualizada, además de que se requieren agregar nuevas funcionalidades.

Por todo lo anteriormente expuesto se define como **problema científico a resolver** ¿Cómo gestionar de forma centralizada la información de Recursos Humanos en la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación?

El **objeto de estudio** de la investigación se centra en los sistemas para la gestión de información de Recursos Humanos, teniendo como **campo de acción**: gestión de la información de Recursos Humanos en la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación.

Para darle solución al problema antes mencionado se propone como **objetivo general**: Elaborar un sistema que centralice la información de Recursos Humanos en la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación.

Se identificaron los siguientes **objetivos específicos**:

- 1- Realizar un estudio del estado del arte de las diferentes tecnologías existentes referente al tema.
- 2- Determinar las funcionalidades que debe cumplir el sistema.
- 3- Describir la propuesta del sistema.
- 4- Implementar la solución propuesta.
- 5- Probar el producto desarrollado.

Se sustenta como **idea a defender**: la planificación, diseño e implementación de un sistema de gestión de información de Recursos Humanos, posibilitará centralizar la información de Recursos Humanos en la Dirección de Producción Audiovisual del Centro de Tecnologías para la Formación.

Para cumplir con los objetivos y resolver el problema científico se proponen las siguientes **tareas de investigación**:

- 1- Análisis de las herramientas y tecnologías, lenguajes de programación y metodologías de desarrollo de software para la confección de la aplicación.
- 2- Definición de los requerimientos de la aplicación.
- 3- Realización de la planificación y diseño del sistema.
- 4- Estudio de las funcionalidades a implementar.
- 5- Desarrollo de los requisitos solicitados por el cliente.
- 6- Probar el producto final.

Métodos científicos utilizados en la investigación:

Métodos Teóricos

Analítico-Sintético: Se utilizó este método con el objetivo de buscar todo lo referente a los sistemas de gestión de información, pues da la posibilidad de analizar las teorías y documentos relacionados, permitiendo de esa forma extraer los elementos más importantes y significativos sobre el tema.

Histórico-Lógico: Este método permite constatar teóricamente cómo han evolucionado los sistemas de gestión de información, pues permite estudiar de forma analítica la trayectoria, evolución y desarrollo.

Métodos Empíricos.

Entrevista: Se realizaron entrevistas a los clientes que se beneficiarán con el sistema, con el objetivo de identificar sus necesidades y determinar las principales funcionalidades de la aplicación.

Observación: Posibilita observar la evolución del trabajo realizado, atentos a posibles atrasos o fallas en el cumplimiento de las tareas propuestas. Además este método permitirá recoger información de todos y cada uno de los indicadores de las variables que figuran en el problema.

El presente trabajo está constituido por 3 capítulos, los cuales se presentan a continuación:

Capítulo 1: Fundamentación Teórica. En este capítulo se resume el estudio realizado sobre los sistemas de gestión de información de Recursos Humanos existentes tanto a nivel internacional, nacional y en la Universidad de las Ciencias Informáticas, además se aborda sobre las metodologías de desarrollo de

software, los lenguajes de programación y de modelado, así como otras herramientas y tecnologías a utilizar en el desarrollo del sistema.

Capítulo 2: Características del Sistema. Se plantea una propuesta de solución donde se aborda todo lo referente a la fase de exploración, planificación y diseño del sistema, requisitos funcionales, así como la descripción del sistema en historias de usuario para su posterior implementación.

Capítulo 3: Implementación y Pruebas. En este capítulo se expone todo lo relacionado con la implementación y las pruebas de aceptación al software, para de esta manera evaluar su funcionamiento de acuerdo a las historias de usuario.

Capítulo 1

Fundamentación Teórica

1.1 Introducción

Con el auge de las tecnologías de la información y las telecomunicaciones se ha posibilitado la automatización de procesos en todos los campos de la vida humana, donde la información y el factor humano son la base central de cualquier estructura empresarial. Los sistemas de gestión de información, específicamente referentes a Recursos Humanos, permiten un control y seguimiento de todo el personal de una institución, lo que posibilita un mayor crecimiento económico y optimización de los diferentes procesos.

1.2 Conceptos asociados al problema de investigación

1.2.1 Sistemas de gestión de información.

La gestión de información está definida como el proceso que se encarga de gestionar la información necesaria para la toma de decisiones y un mejor funcionamiento de los procesos, productos y servicios de una organización. La correcta gestión de información conoce, incorpora y vincula todos los tipos de datos, de todas las áreas de la organización y se relaciona con todos los procesos, desde la generación de datos internos, la selección y adquisición de documentos hasta la organización en su uso. (1)

1.2.2 Gestión de Recursos Humanos

La gestión de Recursos Humanos consiste en el proceso de planear, reclutar, seleccionar, capacitar, desarrollar, motivar, estimular, controlar y evaluar el desempeño del potencial humano de la organización con vistas al logro de sus objetivos. (2)

1.3 Estudio de sistemas similares.

Todas las empresas necesitan gestionar la información de los empleados de forma eficaz, a continuación se realiza un estudio de varios sistemas que permiten automatizar la gestión de información de Recursos Humanos en diferentes ámbitos.

1.3.1 Sistemas de Gestión de Recursos Humanos a nivel internacional.

Microsoft Dynamics ERP: Es una línea de soluciones adaptables para la gestión empresarial que permite a los empleados tomar decisiones empresariales importantes con un mayor grado de confianza, facilita la adopción y la reducción de los riesgos inherentes a la implantación de una nueva solución. Estas soluciones automatizan y optimizan los procesos de planificación de recursos empresariales y gestión de relaciones con el cliente de tal manera que pueden ayudarle a conseguir el éxito empresarial.

Las soluciones de gestión de Recursos Humanos de Microsoft Dynamics pueden ayudarle a sacarle el máximo rendimiento al potencial de los empleados y a aumentar su lealtad, al tiempo que se reducen los costos y la complejidad de la gestión de nóminas, las prestaciones, etc.

Microsoft Dynamics está formado por varios productos de software exclusivos creados para satisfacer prácticamente cualquier necesidad empresarial, tales como: Microsoft Dynamics AX, Microsoft Dynamics CRM, Microsoft Dynamics NAV, etc. (3)

HUM&SOFTWARE: Es un sistema que ha sido especialmente diseñado para satisfacer todas las necesidades del proceso de reclutamiento, selección y evaluación de personal en las empresas. Este sistema proporciona una solución práctica y eficiente a las tareas de:

Bolsa de trabajo:

1. Archivo electrónico de expedientes.
2. Auto-Registro de candidatos vía Internet.
3. Localización de candidatos apegados a perfiles precisos.

Automatización del proceso de reclutamiento y selección:

1. Control total del personal.
2. Seguimiento a detalle del proceso de cada candidato.
3. Retroalimentación instantánea a los candidatos vía e-mail. (4)

GesRRHH 2.0: Permite conservar y gestionar la información relativa a empleados. La herramienta se compone de diferentes módulos: nóminas, plantillas, actividades, agenda y evaluación de desempeño. Cada módulo tiene un color diferente para que sea fácil su identificación. Está diseñada pensando en el usuario, por eso su interfaz es intuitiva y manejable. (5)

El análisis de los sistemas antes mencionados, sirve de guía en el seguimiento de aquellos que muestran las mejores prácticas adaptables a las necesidades de nuestro producto. No obstante no cumplen con los requerimientos en cuanto a costo y funcionalidad, ya que Microsoft Dynamics ERP y HUM&SOFTWARE no se pueden adquirir por el alto costo de su licencia, mientras que la aplicación GesRRHH 2.0 es libre pero no soporta funcionalidades específicas necesarias para el sistema a desarrollar como por ejemplo: exportar acta de admisión, acta de advertencia, modelo de baja, certificado médico, evaluaciones de desempeño a formato (pdf), tomar asistencia, enviar correos, posibilidad de publicar trabajos, documentos y planillas.

1.3.2 Sistema de Gestión de Recursos Humanos a nivel nacional.

Software RH-CITMA versión 1.0: El software pretende obtener las ventajas de la identificación de los procesos de la empresa para la gestión de algunas actividades de Recursos Humanos. Para lograr lo anterior ha sido diseñado como sistema compuesto por diferentes módulos relacionados entre sí dentro de los que se encuentran:

1. Diseño de empresa.
2. Procesos.
3. Profesiogramas.
4. Selección del personal.
5. Peligros en el puesto.
6. Accidentes laborales.
7. Sistema de sugerencias.
8. Diagnóstico de clima organizacional.
9. Formación.
10. Inventario de personal.

Este software ha sido diseñado desde el primer momento para convertirse en versiones posteriores en un sistema para la gestión de la calidad o incluso para la gestión integral que incluya la seguridad, salud del trabajo y el medio ambiente. (6)

El sistema RH-CITMA posee valores positivos a tomar para la implementación del sistema, pero de igual manera no cumple en su totalidad con las características del sistema a desarrollar ya que no presenta funcionalidades para controlar la asistencia, capacitación, estudios, sanciones de los trabajadores, publicar trabajos, documentos, planillas y envío de correo electrónico.

1.3.3 Sistemas de Gestión de Recursos Humanos en la UCI.

En la Universidad existen varios sistemas para gestionar información referente al personal que trabaja en dicha institución de altos estudios, alguno de estos son:

Sistema de Registro Integral (ASSETS): Sistema que registra todo el personal que es plantilla en la UCI, es del tipo de Planificación de Recursos Empresariales (ERP, por sus siglas en ingles), de factura foránea. Se introdujo en Cuba en el año 1997. Controla plenamente los recursos laborales, dígame: empleados, estructura organizativa de la entidad y plantilla. Es válido aclarar que este sistema no se desarrolló en la UCI, sino que se modificó teniendo en cuenta el personal que se gestiona en la universidad, proceso propio de la misma. La aplicación consta de diferentes módulos, como finanzas, control de inventario, activos fijos, recursos humanos, nómina, entre otros.

Este sistema no puede emplearse por la Dirección de Producción Audiovisual (DPA) del Centro FORTES ya que el mismo por razones de seguridad y conexiones limitadas a la base de datos central, solo tienen acceso los directivos y empleados del Departamento de Recursos Humanos de la UCI, además esta aplicación no permite controlar asistencia, llevar la continuidad de la capacitación y estudios universitarios de los trabajadores, no posibilita gestionar las evaluaciones de desempeño, estimulaciones morales y no permite exportar a formato (pdf) modelos de evaluación, asistencia, entre otros.

Sistema Integral de Gestión - CEDRUX: Aplicación web todavía en implementación, la cual está siendo desarrollada en la UCI, la primera solución a desplegar es la base para el desarrollo del ERP-Cuba y debe contener las funcionalidades de los softwares nacionales certificados más utilizados en el país. Esta aplicación está compuesta por módulos, los cuales son utilizados en dependencia de las necesidades de cada empresa, uno de ellos es el subsistema de capital humano, entre sus funcionalidades implementadas están:

1. Definir plantilla de cargo.

2. Definir los puestos de trabajo.
3. Registro de persona.
4. Formalizar relación laboral.
5. Movimiento de la fuerza de trabajo.
6. Utilización del fondo de tiempo laboral.
7. Vacaciones anuales pagadas.
8. Seguridad social.
9. Disciplina laboral.
10. Nómina.
11. Retenciones.

El sistema CEDRUX, específicamente el subsistema de capital humano no puede utilizarse por la DPA, ya que el mismo se encuentra en desarrollo.

DPARH: Aplicación actual que utiliza la DPA del Centro FORTES, software de escritorio desarrollado en C# con el IDE SharpDevelop, entre sus limitantes se encuentra el hecho de que la información que maneja no está centralizada, por lo que no se encuentra actualizada en cada una de las máquinas en las cuales está instalado el software, no dispone de nuevas funcionalidades que necesita la DPA, además de no ser multiplataforma, por las características antes mencionadas de esta aplicación, no es factible para la entidad, por lo que se hace necesario el desarrollo de un nuevo sistema.

1.4 Herramientas y tecnologías informáticas.

A continuación se describen las herramientas y tecnologías que serán utilizadas para darle solución al problema, teniendo en cuenta las necesidades existentes y el entorno de trabajo donde se aplicará la propuesta.

1.4.1 Metodologías para el desarrollo de software.

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con calidad. Define las etapas para el desarrollo, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto. (7)

Metodología Tradicional

Las metodologías robustas o tradicionales están pensadas para desarrollar soluciones empresariales complejas o de largo alcance donde se dificulta la comunicación y sincronización entre los miembros del equipo de desarrollo. Éstas imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un producto terminado más eficiente y predecible.

Rational Unified Process (Proceso Unificado de Desarrollo, RUP)

RUP es uno de los procesos más generales de los existentes, ya que está pensado para adaptarse a cualquier proyecto.

Un proyecto realizado siguiendo la metodología RUP se divide en cuatro fases:

1. Inicio (puesta en marcha).
2. Elaboración (definición, análisis, diseño).
3. Construcción (implementación).
4. Transición (fin del proyecto y puesta en producción).

Fases de RUP

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.

RUP define nueve actividades a realizar en cada fase del proyecto:

1. Modelado del negocio.
2. Análisis de requisitos.
3. Análisis y diseño.
4. Implementación.
5. Pruebas.
6. Distribución.
7. Gestión de configuración y cambios.
8. Gestión de proyecto.

9. Gestión del entorno.

Se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, se documenta lo mejor posible y utiliza el lenguaje unificado de modelado (UML). (8)

Metodologías Ágiles.

Las metodologías ágiles están especialmente orientadas a proyectos con requisitos poco definidos o cambiantes, se aplican bien en equipos pequeños que resuelven problemas concretos, lo que no dificulta su aplicación en el desarrollo de grandes sistemas, ya que una correcta modularización de los mismos es fundamental para su exitosa implantación. Dividir el trabajo en módulos abordables minimiza los fallos y el costo. Estas metodologías presentan diversas ventajas, entre las que podemos destacar:

1. Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo
2. Entrega continua y en plazos breves de software funcional
3. Trabajo conjunto entre el cliente y el equipo de desarrollo
4. Importancia de la simplicidad, eliminando el trabajo innecesario
5. Atención a la excelencia técnica y al buen diseño
6. Mejora continua de los procesos y el equipo de desarrollo. (9)

Extreme Programming (Programación Extrema, XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

XP propone un conjunto de reglas y prácticas que se listan a continuación:

1. Juego de la planificación.
2. Entregas pequeñas.
3. Metáfora.
4. Diseño simple.
5. Pruebas.
6. Refactorización (Refactoring).
7. Programación en parejas.
8. Propiedad colectiva del código.
9. Integración progresiva.
10. 40 horas por semana.
11. Cliente in-situ.
12. Estándares de programación.

SCRUM

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, y obtener el mejor resultado posible de un proyecto. Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados inmediatos, los requisitos son cambiantes o poco definidos, y la innovación, competitividad, flexibilidad, productividad son fundamentales. (10)

Características

1. La incertidumbre como elemento consustancial y asumido en el entorno y en la cultura de la organización.
2. Equipos de desarrollo auto-organizados.
3. Fases de desarrollo solapadas.
4. Control sutil.
5. Difusión y transferencia del conocimiento. (11)

Beneficios

1. Cumplimiento de expectativas: El cliente establece sus expectativas indicando el valor que le aporta cada requisito / historia del proyecto, el equipo los estima y con esta información el cliente establece su prioridad.
2. Flexibilidad a cambios: Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.

3. Reducción del tiempo de mercado: El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
4. Mayor calidad del software: Necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
5. Mayor productividad: Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
6. Maximiza el retorno de la inversión (ROI): Producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.
7. Predicciones de tiempos: Mediante esta metodología se conoce la velocidad media del equipo por sprint.
8. Reducción de riesgos: El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada. (12)

Una vez estudiadas las posibles metodologías a utilizar, se decide el uso de XP ya que esta metodología es capaz de adaptarse a los cambios de requisitos que puedan ocurrir en el transcurso de vida del proyecto, entre otras ventajas pueden mencionarse los pocos requerimientos de documentación y planificación, es concebida para ser utilizada en proyectos pequeños, así como la exigencia de tener siempre el cliente disponible para el desarrollo.

1.4.2 Lenguaje de modelado

UML (Lenguaje unificado de modelado)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Grupo de Gestión de Objetos). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (13)

1.4.3 Herramientas CASE (Ingeniería de software asistida por computadora)

Visual Paradigm 6.4

Es una herramienta CASE, la misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm soporta el ciclo de vida completo del proceso de desarrollo de software a través de la representación de todo tipo de diagramas. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos.

Algunas de sus características son:

1. Disponibilidad en múltiples plataformas.
2. Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
3. Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
4. Capacidades de ingeniería directa e inversa.
5. Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
6. Disponibilidad de múltiples versiones, para cada necesidad.
7. Licencia: gratuita y comercial.
8. Soporta aplicaciones Web.

Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software (14).

Rational Rose

Es una herramienta CASE, desarrollada por Rational Corporation basada en UML, que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del software.

Características adicionales incluidas:

1. Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
2. Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
3. La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
4. Capacidad de análisis de calidad de código.
5. El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones Web.
6. Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
7. Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
8. Publicación web y generación de informes para optimizar la comunicación dentro del equipo. (15)

Se selecciona la herramienta Visual Paradigm ya que posibilita la interoperabilidad entre diagramas, es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, además de permitir la transformación de diagramas de Entidad-Relación en tablas de base de datos, también realiza ingeniería inversa a bases de datos desde sistemas gestores de bases de datos existentes a diagramas de Entidad-Relación.

1.4.4 Lenguajes de Programación

Un lenguaje de programación es aquel elemento dentro de la Informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que ponen a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. (16)

El programador es el encargado de utilizar un lenguaje de programación para crear un conjunto de instrucciones que, al final, constituirá un programa o subprograma informático. (16)

Los lenguajes de programación pueden clasificarse según el paradigma que usan en: procedimentales, orientados a objetos, funcionales, lógicos, híbridos, entre otros. (16)

Lenguajes de desarrollo del lado del cliente:

Lenguaje JavaScript

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas, éstas incorporan efectos, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (17)

JavaScript es dinámico, simple, responde a eventos en tiempo real, permite cambiar parte del diseño de las páginas web, como la información y diversos aspectos sin que estos sean actualizados por el servidor, por lo que hace de la web un lugar más interactivo.

XHTML 1.0

XHTML, acrónimo en inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto). Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. En su versión 1.0, XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium (W3C) de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. (18)

La utilización de XHTML proporciona un mejor rendimiento a las aplicaciones web que existen actualmente, ya que exige una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados por lo que se propone utilizar en el desarrollo del sistema.

CSS 2.0 (Cascading Style Sheets), en español Hojas de estilos en cascadas.

Es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos de la presentación y es imprescindible para crear páginas web complejas. (19)

El W3C define CSS como "un mecanismo simple para añadir estilo, por ejemplo: fuentes, colores, espacios a documentos web". Es un lenguaje de diseño estándar que controla los colores, la tipografía, tamaño, ubicación de elementos e imágenes. (20)

Ventajas de CSS

- 1- Conserva el ancho de banda del usuario, lo que acelera la carga de páginas, especialmente en conexiones telefónicas.
- 2- Reduce la sobrecarga del servidor.
- 3- Reduce el tiempo de diseño y programación.
- 4- Aumenta la accesibilidad ya que elimina algunos, muchos o todos los elementos de presentación del marcado. (20)

Lenguaje de desarrollo del lado del servidor

PHP 5

Es un lenguaje del lado del servidor (esto significa que PHP funciona en un servidor remoto que procesa la página web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas Web dinámicas.

Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades tan extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo. Combinado con la base de datos MySQL, es el lenguaje estándar a la hora de crear sitios de comercio electrónico o páginas Web dinámicas. (21)

Entre sus características fundamentales están:

1. Gratuito: Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación personal o profesional, de manera completamente libre.
2. Gran popularidad: Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
3. Enorme eficiencia: Con escaso mantenimiento y un servidor gratuito, puede soportar sin problema millones de visitas diarias.
4. Sencilla integración con múltiples bases de datos, MySQL es la base de datos que mejor trabaja con PHP, puede conectarse también a PostgreSQL, Oracle o cualquier otra base de datos compatible con ODBC.
5. Versatilidad. PHP puede usarse con la mayoría de sistemas operativos, ya sea basados en UNIX (Linux, FreeBSD), como con Windows, el sistema operativo de Microsoft.
- 6- Gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, PHP está dotado de un gran número de funciones que nos simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con cookies y sesiones, etc. (21)

Se decide utilizar PHP 5 por ser un lenguaje libre, se posee amplio conocimiento del mismo por el desarrollador del sistema, es completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos, presenta gran compatibilidad con MySQL, existe gran cantidad de documentación y cuenta con una biblioteca de funciones sumamente amplia.

1.4.5 IDEs (Entorno de desarrollo integrado)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. (22)

NetBeans 6.9

Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans, es un producto libre y gratuito sin restricciones de uso.

Esta plataforma es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

NetBeans ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

1. Administración de las interfaces de usuario (ej. menús y barras de herramientas).
2. Administración de las configuraciones del usuario.
3. Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
4. Administración de ventanas.
5. Framework basado en asistentes. (23)

Eclipse

Características

1. Soportar las herramientas proporcionadas por diferentes fabricantes de software independientes (ISV's).
2. Soportar herramientas que permitan manipular diferentes contenidos (HTML, Java, C, JSP, EJB, XML, etc.)
3. Facilitar una integración transparente entre todas las herramientas y tipos de contenidos sin tener en cuenta al proveedor.
4. Proporcionar entornos de desarrollo gráfico (GUI) o no gráficos.
5. Ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows y Linux.
6. Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad.

El principal objetivo de la Plataforma Eclipse es proporcionar mecanismos, reglas que puedan ser seguidas por los fabricantes para integrar de manera transparente sus herramientas. Mediante APIs interfaces, clases y métodos, se exponen estos mecanismos. (24)

El IDE seleccionado es el NetBeans en su versión 6.9 ya que el mismo es de código abierto y presenta magníficas cualidades y funciones para el desarrollo web, es multiplataforma, se complementa perfectamente con el framework jQuery, consta de una gran comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una variada selección de plugins de terceros. Además permite la creación de aplicaciones web con la nueva versión de PHP 5.

1.4.6 Frameworks (Marco de trabajo)

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes, proporcionando una estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (25)

Framework del lado del cliente

jQuery 1.7.1

Es un framework JavaScript que ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Posibilita la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de AJAX (JavaScript Asíncrono y XML), además de tener licencia para uso en cualquier tipo de plataforma, personal o comercial. (26)

La gran ventaja de jQuery, es que la página se puede manipular en cuanto se ha cargado su código HTML (y por tanto, se ha construido el árbol DOM de la página) mientras que la función de JavaScript espera a que se carguen todos los elementos de la página, incluyendo todas las imágenes. De esta forma, las aplicaciones realizadas con jQuery pueden responder de forma mucho más rápida que las aplicaciones JavaScript tradicionales. (27)

ExtJS

Es una librería JavaScript ligera y de alto rendimiento destinada a crear entornos de aplicaciones web complejos, compatible con la mayoría de los navegadores, las cuales permiten crear aplicaciones enriquecidas del lado del cliente usando tecnologías como AJAX, DHTML y DOM. Fue creado

originalmente como extensión para Yahoo User Interface (YUI), en la actualidad puede usarse como extensión para las bibliotecas jQuery y Prototype.

Esta librería incluye:

1. Componentes de interfaz de usuario.
2. Modelo de componentes extensibles.
3. Un API fácil de usar.
4. Licencias de código abierto y comercial. (28)

jQuery fue el framework escogido ya que es fácil de usar y aprender, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework, Ostenta una gran comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en jQuery para implementar asuntos como interfaces de usuario y galerías.

Frameworks del lado del servidor

Symfony 1.4

Es un completo Framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones 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 éstas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. Está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas libres como en plataformas Windows. (25)

Características

Symfony se diseñó para que se ajustara a los siguientes requisitos:

1. Fácil de instalar y configurar en la mayoría de las plataformas.

2. Independiente del sistema gestor de bases de datos.
3. Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
4. Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
5. Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
6. Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
7. Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
8. Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros. (25)

CodeIgniter 2.0

CodeIgniter posibilita desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y estructura lógica para acceder a sus librerías, permitiendo creativamente enfocarse en su proyecto, minimizando la cantidad de código necesario para una tarea dada. (29)

Características:

1. Libre: CodeIgniter se encuentra bajo una licencia Open Source.
2. Liviano: El núcleo del sistema sólo requiere unas pocas pequeñas librerías. A diferencia de muchos entornos de trabajo que requieren significativamente más recursos. Las librerías adicionales son cargadas dinámicamente a pedido, basado en sus necesidades para un proceso dado, así que el sistema base es muy delgado y bastante rápido.
3. Soporta a PHP 5.
4. Usa Modelo Vista Controlador: Permitiendo una buena separación entre lógica y presentación.
5. Genera URLs limpias y amigables a los motores de búsqueda.
6. Paquetes: CodeIgniter viene con un rango lleno de librerías que le permiten realizar las tareas de desarrollo web más comúnmente necesarias, como acceder a una base de datos, mandar un email, validar datos de un formulario, mantener sesiones, manipular imágenes, trabajando con datos XML-RPC y mucho más. (29)

CodeIgniter 2.0 fue el framework escogido ya que es rápido y ligero, existe mucha documentación y es compatible con PHP 5. Características importantes para la aplicación a desarrollar, no se escoge Symfony ya que es ideal para aplicaciones de mayor envergadura por su gran robustez, por lo que no sería idóneo para un proyecto pequeño.

1.4.7 Servidores Web

Un servidor web programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). (30)

Apache 2

Es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada.

Ventajas y aceptación

1. Modular.
2. Multi-plataforma.
3. Extensible.
4. Popular (fácil conseguir ayuda/suporte). (30)

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo.

IIS (Internet Information Services)

Es un potente servidor web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones web sobre todas las versiones de Windows Server 2003. IIS hace posible que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones Web y a la vez reducir sus

costes administrativos. IIS 6.0 soporta la Iniciativa de Sistemas Dinámicos de Microsoft (DSI) con aislamiento de procesos y capacidades de gestión mejoradas. (31)

Se empleara en la solución el servidor web Apache por todas las características antes mencionadas, no se empleará IIS ya que es de la empresa Microsoft Corporation y es privativo, lo que constituye una gran limitante para emplearlo en la solución.

1.4.8 Gestores de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) 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. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. (32)

MySQL 5.1

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. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.

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 como Yahoo!, Finance, Google, Motorola, entre otras. (33)

Es, sin duda, la base de datos más popular y utilizada a la hora de desarrollar páginas web dinámicas y sitios de comercio electrónico. Se suele trabajar en combinación con PHP, y comparte con éste algunas de las características que lo convierten en una elección segura. Entre ellas:

1. Gratuito: Puede ser utilizado sin limitación alguna. Se encuentra bajo licencia GPL.
2. Popularidad: Son innumerables las páginas donde encontrar información.
3. Rapidez: La velocidad de proceso de MySQL es legendaria.
4. Versatilidad: Trabaja tanto con sistemas operativos basados en Unix como con el sistema operativo Windows, de Microsoft.
5. Sencillez de manejo: Al utilizar el lenguaje estándar SQL, el tener conocimientos de otras bases de datos nos ayudará enormemente. (32)

PostgreSQL

Es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, de código abierto, brinda un control de concurrencia multi-versión 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.

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.

El código fuente se encuentra disponible para todos sin costo alguno. Está disponible para 34 plataformas con la última versión estable.

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. (32)

Después de realizar un estudio sobre los dos gestores de base de datos más reconocidos se escoge MySQL ya que tiene una mayor velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrece mayor rendimiento, su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema, las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración.

1.5 Conclusiones del capítulo

En este capítulo se ha analizado el estado del arte de las herramientas existentes que se relacionan con el objeto de estudio de la investigación, estudiando los diferentes lenguajes de programación web, herramientas y metodologías posibles a utilizar, llegando a la conclusión de utilizar una metodología de desarrollo ágil, en este caso, XP, pues está probada su eficiencia en equipos de desarrollo pequeños, donde las entregas frecuentes, las iteraciones cortas son premisas que ayudarán a entregar un software funcional y probado tempranamente.

También se puede concluir que la utilización del Visual Paradigm 6.4 como herramienta CASE, que emplea UML como lenguaje de modelado, además de ser libre y multiplataforma unido al servidor web Apache 2, el IDE NetBeans 6.9, MySQL 5.1 y los Frameworks CodeIgniter 2.0 y JQuery 1.7.1 son la opción más conveniente para el desarrollo del sistema. Además de los lenguajes seleccionados se encuentran JavaScript, XHTML 1.0, CCS y PHP 5, los cuales sin duda alguna han constituido un buen punto de partida para el desarrollo final.

Capítulo 2

Características del Sistema

2.1 Introducción:

Desarrollar un producto informático con calidad no es tarea fácil, por lo cual es importante e imprescindible la utilización de metodologías de desarrollo de software, ya que estas brindan un conjunto de pasos y procedimientos que deben seguirse para lograr el éxito final. Entre las metodologías tradicionales y ágiles, las segundas están especialmente orientadas a proyectos pequeños y con requerimientos imprecisos. La programación extrema como ejemplo de éstas, propone un conjunto de reglas y prácticas a seguir para lograr de forma rápida y sin generar una documentación excesiva automatizar todos los requisitos funcionales previstos por el cliente. A continuación se detallan las fases de Planificación, Diseño y los artefactos generados en cada fase.

2.2 Objeto de automatización.

1. Autenticar.
2. Gestionar cuentas en el sistema.
3. Gestionar trabajadores.
4. Gestionar evaluación de desempeño, capacitación, estudios, sanciones, estimulación moral y salario de los trabajadores.
5. Exportar modelos.
 - 5.1. Exportar modelo de certificado médico.
 - 5.2. Exportar modelo de asistencia.
 - 5.3. Exportar modelo de evaluación de desempeño.
 - 5.4. Exportar certificado de vinculación para docentes.
 - 5.5. Exportar certificado de evaluación de desempeño.
6. Realizar búsquedas filtradas.
 - 6.1. Conocer trabajadores que estudian en la educación superior.
 - 6.2. Conocer trabajadores sancionados.
 - 6.3. Conocer trabajadores en capacitación.

- 6.4. Conocer trabajadores según evaluación de desempeño.
- 6.5. Conocer trabajadores según salario.
- 6.6. Conocer trabajadores según estimulación moral.
- 6.7. Conocer trabajadores según cargo.
- 6.8. Conocer trabajadores según área.
- 6.9. Conocer trabajadores según asistencia.
- 7. Controlar asistencia diaria.
 - 7.1. Registrar hora de entrada y salida de la mañana de un trabajador.
 - 7.2. Registrar hora de entrada y salida de la tarde de un trabajador.
- 8. Exportar actas.
 - 8.1. Exportar acta de advertencia.
 - 8.2. Exportar acta de amonestación.
 - 8.3. Exportar acta de comité de ingreso.
 - 8.4. Exportar acta de admisión.
 - 8.5. Exportar acta de boleta de admisión.
- 9. Envío de correo electrónico a los trabajadores.
- 10. Publicación de trabajos, informes y planillas.

2.3 Información que se maneja.

Se procesan datos referentes a los Recursos Humanos de la DPA del Centro FORTES.

2.4 Propuesta de sistema.

Se propone la implementación de una aplicación web utilizando como estructura de soporte el framework CodeIgniter integrado con el servidor web apache que será capaz de gestionar información referente a los Recursos Humanos en la DPA del Centro FORTES. El sistema tendrá una autenticación por roles el cual permitirá realizar operaciones en dependencia del tipo de usuario autenticado permitiendo la seguridad de los datos que se manejan. Posibilitará la opción de publicar trabajos, informes, planillas y envío de correo electrónico a los trabajadores.

2.5 Personas relacionadas con el sistema.

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de aquellas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Tabla 1: Personas relacionadas con el sistema

Personas relacionadas con el sistema	Justificación
Administrador	Dispone de posibilidades ilimitadas para ejecutar todas las funciones del sistema.
Usuario	Dispone de posibilidades limitadas, solo podrá ver información existente en el sistema.

2.6 Exploración

La metodología de desarrollo Programación Extrema (XP) comienza con su fase de exploración. Es la etapa del proceso de desarrollo de software que propone XP para comenzar la construcción de un producto. Una vez que los clientes entregan su propuesta al equipo de trabajo, comienza el análisis en grupo. Un aspecto clave en el esclarecimiento de las dudas sobre los procesos a automatizar es que, desde el mismo inicio del proyecto, un miembro del equipo del trabajo del cliente, se adiciona al de desarrolladores.

2.6.1 Lista de Reserva del Producto (LRP).

La LRP es una lista priorizada que define el trabajo a realizar en el proyecto. En ella se incluyen los requerimientos del sistema, por lo que puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. El completamiento de la lista, acompañado de los cambios en el entorno y el producto asegura que el producto sea lo más correcto, útil y competitivo posible.

Se describen los requisitos funcionales ordenados teniendo en cuenta la prioridad de implementación (Alta, Media y Baja), además de mencionar los requisitos no funcionales del sistema.

Incluye los siguientes campos: **Prioridad**, que contiene los niveles de prioridad de las funcionalidades incluidas (divididas en 3 grupos); **Ítem**, que contiene la numeración secuencial y **Descripción**, que contiene la descripción de la funcionalidad. Además, se incluyen los **Requerimientos No Funcionales** (que no llevan estimación).

Tabla 2: Lista de Reserva del Producto (LRP)

Prioridad	Ítem *	Descripción
Alta		
Autenticar.		
	1	Solicitar usuario y contraseña.
	2	Validar en la base de datos de acuerdo con el usuario a autenticar.
Gestionar cuentas.		
	3	Crear cuentas.
	4	Modificar cuentas.
	5	Eliminar cuentas.
	6	Mostrar cuentas.
Gestionar trabajador.		
	7	Insertar trabajador
	8	Modificar trabajador.
	9	Eliminar trabajador.
	10	Mostrar datos de un trabajador.
Controlar asistencia diariamente.		
	11	Registrar hora de entrada y salida de la mañana de un trabajador.
	12	Registrar hora de entrada y salida de la tarde de un trabajador.
	13	Exportar modelo de asistencia.
Gestionar salario.		
	14	Insertar salario.
	15	Modificar salario.

	16	Eliminar salario.
	17	Mostrar salario.
Gestionar evaluación de desempeño.		
	18	Insertar evaluación.
	19	Modificar evaluación.
	20	Eliminar evaluación.
	21	Mostrar evaluación.
Realizar búsquedas filtradas.		
	22	Conocer trabajadores según estudios en la educación superior.
	23	Conocer cantidad de trabajadores según cargo.
	24	Conocer cantidad de trabajadores según evaluación de desempeño.
	25	Conocer trabajadores sancionados.
	26	Conocer trabajadores según estimulaciones morales.
	27	Conocer trabajadores según capacitaciones.
	28	Conocer trabajadores según salario.
	29	Conocer cantidad de trabajadores según área.
	30	Conocer trabajadores según asistencia.
Media		
Exportar modelos.		
	31	Evaluación de desempeño.
	32	Certificado de evaluación de desempeño.
	33	Certificado médico.
	34	Certificado de vinculación para docentes.
	35	Asistencia.
Bajas		
Exportar actas.		
	36	Admisión.
	37	Boleta de admisión

	38	Advertencia.
	39	Comité de ingreso.
	40	Amonestación.
Gestionar capacitación.		
	41	Insertar capacitación
	42	Modificar capacitación
	43	Eliminar capacitación
	44	Mostrar capacitación
Gestionar estudios.		
	45	Insertar estudios.
	46	Modificar estudios.
	47	Eliminar estudios.
	48	Mostrar estudios.
Gestionar sanciones.		
	49	Insertar sanciones.
	50	Modificar sanciones.
	51	Eliminar sanciones.
	52	Mostrar sanciones.
Gestionar estimulación moral.		
	53	Insertar estimulación moral.
	54	Modificar estimulación moral.
	55	Eliminar estimulación moral.
	56	Mostrar estimulación moral.
Envío de correo electrónico a los trabajadores.		
Publicación de trabajos, informes y planillas.		

Requisitos mínimos necesarios del sistema.

Requisitos de Software.

Se debe disponer para poder instalar la aplicación:

En el cliente

Sistema operativo Linux o Windows (98 ó superior).

Navegador web estándar con capacidad de interpretación de JavaScript y CSS compatible con Internet Explorer 7 o superior, Netscape, Mozilla 3.5 o superior.

En el servidor

Servidor Web Apache.

PHP 5 o superior como lenguaje de programación.

MySQL 5 o superior como servidor de base de datos.

Requisitos de Hardware:

Las PC servidoras deberán contar con un microprocesador Intel Pentium III o superior, a 450MHz o superior, con un mínimo de 192MB de memoria RAM, un disco duro con capacidad de 1GB o superior y una tarjeta de red alámbrica, inalámbrica o módem.

2.6.2 Historias de usuario

Historias de usuario del inglés User Stories (en adelante HU), son la forma en que se especifican en XP los requisitos del sistema. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Durante este proceso se identificaron 15 historias de usuario, las cuales se describen a continuación:

Tabla 3 HU Autenticar usuario.

Historia de Usuario	
No: 01	Nombre historia de usuario: Autenticar usuario.
Usuario: Administrador y usuarios.	Iteración Asignada: Primera.
Prioridad: Alta.	Puntos Estimados: 1 Semana.
Riesgo en Desarrollo: Media.	Puntos Reales: 1 Semana.
Descripción: El usuario introduce sus datos en el sistema y este verifica los mismos y automáticamente el usuario es autenticado en el sistema.	
Observaciones:	

Tabla 4 HU Gestionar cuentas.

Historia de Usuario	
No: 02	Nombre historia de usuario: Gestionar cuentas.
Usuario: Administrador.	Iteración Asignada: Primera.
Prioridad: Alta.	Puntos Estimados: 1 Semana.
Riesgo en Desarrollo: Media.	Puntos Reales: 1 Semana.
Descripción: El administrador podrá crear, modificar, buscar y eliminar una cuenta del sistema.	
Observaciones:	

Tabla 5 HU Gestionar trabajadores.

Historia de Usuario	
No: 03	Nombre historia de usuario: Gestionar trabajadores
Usuario: Administrador.	Iteración Asignada: Primera.
Prioridad: Alta.	Puntos Estimados: 2 Semana.
Riesgo en Desarrollo: Media.	Puntos Reales: 2 Semana.
Descripción: El administrador podrá crear, modificar, buscar y eliminar un trabajador.	
Observaciones:	

Tabla 6 HU Realizar búsquedas filtradas.

Historia de Usuario	
No: 04	Nombre historia de usuario: Realizar búsquedas filtradas.
Usuario: Administrador u usuarios.	Iteración Asignada: Primera.
Prioridad: Alta.	Puntos Estimados: 1 Semana.
Riesgo en Desarrollo: Media.	Puntos Reales: 1 Semana.
Descripción: Permitirá realizar búsquedas de acuerdo a la información que se necesite.	
Observaciones:	

Ver restantes Historias de usuario (Anexo 1).

2.7 Planificación

XP plantea la planificación como un permanente diálogo entre el cliente y los desarrolladores del proyecto, en la que los primeros decidirán el alcance, qué es lo realmente necesario del proyecto, la prioridad con qué debe ser desarrollado en primer lugar. En cuanto a los desarrolladores, son los responsables de estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las consecuencias de determinadas decisiones, de organizar la cultura de trabajo y, finalmente, de realizar la planificación detallada dentro de cada versión.

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar todas las historias de usuario siendo uno de los artefactos generados en esta fase. Una vez terminado esto, se procede a organizarlas en las iteraciones correspondientes, en dependencia de la prioridad especificada por el cliente y del tiempo de desarrollo de cada una.

Estimación de esfuerzos por historias de usuario.

Tabla 7 Estimación de esfuerzos por historias de usuario

Historias de Usuarios	Estimación (semanas)
Autenticar	1

Gestionar cuentas.	1
Gestionar trabajador.	2
Controlar asistencia diariamente.	1
Gestionar salario.	1
Gestionar evaluación de desempeño.	1
Realizar búsquedas filtradas.	1
Exportar modelos.	2
Exportar actas.	1
Controlar capacitación.	1
Gestionar estudios.	1
Gestionar sanciones.	1
Gestionar estimulación moral.	1
Envío de correo electrónico a los trabajadores.	1
Publicación de trabajos, informes y planillas.	1

2.7.1 Plan de iteraciones.

Después de haber identificado las historias de usuarios y realizado una previa estimación de esfuerzos se procede a la planificación de la etapa de implementación del sistema. En este plan se especifica la prioridad con que se irán implementando las historias de usuarios organizadas por iteraciones así como posibles fechas de liberación. Este paso es de suma importancia en el ciclo de desarrollo de XP. Las funcionalidades son planificadas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración.

Una iteración no es más que un mini-proyecto en el que se obtiene un resultado en software con un valor para el cliente. Claro está que no se tendría la versión completa del producto, ya que este concluye al finalizar la última iteración.

Iteración 1.

Tiene como objetivo la implementación de las historias de usuario de mayor prioridad. Al finalizar se contará con las funcionalidades descritas en las historias de usuario asignadas.

Iteración 2.

El objetivo de esta iteración es la implementación de las restantes funcionalidades con prioridad alta, además, de otras con prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las historias de usuario asignadas.

Iteración 3.

En esta iteración serán implementadas las funcionalidades de prioridad baja.

2.7.2 Plan de duración de las iteraciones.

Siguiendo el desarrollo de la metodología XP se crea el plan de duración de las iteraciones. En este plan se especifica más detalladamente el orden de desarrollo de las historias de usuarios dentro de cada iteración así como la estimación completa de dicha iteración.

Tabla 8 Plan de duración de las iteraciones.

Iteraciones	Orden de las Historias de Usuarios a implementar.	Duración de las iteraciones.
Iteración 1	Autenticar. Gestionar cuentas. Gestionar trabajador. Realizar búsquedas filtradas.	5 semanas
Iteración 2	Controlar asistencia diariamente. Gestionar salario. Gestionar evaluación de desempeño. Exportar modelos.	5 semanas
Iteración 3	Exportar actas. Gestionar capacitación. Gestionar estudios. Gestionar sanciones. Gestionar estimulación moral. Envío de correo electrónico a los trabajadores. Publicación de trabajos, informes y planillas.	7 semanas

2.7.3 Plan de entregas

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 (clientes, gerentes, desarrolladores). Este plan se acopla las funcionalidades referentes a un mismo tema en módulos, esto permite un mayor entendimiento en la fase de implementación.

Tabla 9 Plan de entregas.

Historia de usuario	Fin 1ra Iteración (1ra Semana de marzo).	Fin 2da Iteración (1ra Semana de abril).	Fin 3da Iteración (3ra Semana de abril).
Autenticar.	V 1.0	finalizado	finalizado
Gestionar cuentas en el sistema.	V 1.0	finalizado	finalizado
Gestionar trabajador	V 1.0	finalizado	finalizado
Realizar búsquedas filtradas.	V 1.0	finalizado	finalizado
Controlar asistencia		V 1.0	finalizado
Gestionar salario.		V 1.0	finalizado
Gestionar evaluación de desempeño.		V 1.0	finalizado
Exportar modelos		V 1.0	finalizado
Exportar actas			V 1.0
Gestionar capacitación.			V 1.0
Gestionar estudios.			V 1.0
Gestionar sanciones.			V 1.0
Gestionar estimulación moral.			V 1.0
Envío de correo electrónico a los trabajadores.			V 1.0
Publicación de trabajos, informes y planillas.			V 1.0

2.8 Diseño del Sistema

2.8.1 Arquitectura del sistema

Entre los patrones arquitectónicos empleados en la construcción de aplicaciones web se encuentra, el Modelo - Vista - Controlador (MVC), es un patrón de desarrollo o un estilo de arquitectura de software que separa el código fuente de las aplicaciones en tres grupos:

Modelo: Se mantendrá encapsulada la complejidad de la base de datos y simplemente se crearan funciones para recibir, insertar, actualizar o borrar información de nuestras tablas.

Vista: Codifica y mantiene la presentación final de nuestra aplicación de cara al usuario, contiene todo el código HTML, CSS, JavaScript, entre otros; que se tiene que generar para producir una determinada página.

Controlador: Sirve de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web.

Particularmente el framework CodeIgniter utiliza como arquitectura base el patrón MVC.

Algunas de las ventajas de este patrón son las siguientes:

1- Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no existe una dependencia directa entre ambos, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación web pueden utilizar el mismo modelo de objetos mostrado de modos diferentes.

2- Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas del negocio. Dado que el modelo no depende de la vista, agregar nuevas opciones de presentación generalmente no afecta al mismo.

2.8.2 Patrones de Diseño

Un patrón de diseño define un esquema de refinamiento de los subsistemas o componentes dentro de un sistema, o las relaciones entre estos. Este describe una estructura común y recurrente de componentes interrelacionados, que resuelve un problema general de diseño dentro de un contexto particular.

Los patrones de diseño trabajan a una escala intermedia. Son menores en escala que un patrón de arquitectura, pero logran ser independientes del lenguaje de programación. La aplicación de un patrón de

diseño no tiene efectos sobre la estructura fundamental del sistema (arquitectura), pero puede tener una fuerte influencia sobre la arquitectura de un subsistema. (34)

Patrones GRASP

GRASP es el acrónimo de General Responsibility Assignment Software Patterns (Patrones generales de software para asignar responsabilidades), nombre elegido para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos. (35)

Alta cohesión: Una clase con alta cohesión posee un número relativamente pequeño de responsabilidades, relacionadas entre sí por sus funcionalidades. Dicha clase colabora con otras clases para compartir el esfuerzo si la tarea es grande. Este patrón mejora la calidad y facilidad del diseño, genera un bajo acoplamiento y promueve la reutilización. (35)

Controlador: La mayor parte de los sistemas reciben eventos de entrada externa. En estos casos hay que elegir controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas en la elección de los controladores de eventos. Su utilización propicia que las operaciones del sistema se manejen en la capa de dominio de los objetos, y no en la de presentación. (35)

Experto: Permite la asignación de una responsabilidad a la clase que cuenta con la información necesaria para cumplirla (experto de información). Expresa la "intuición" de que los objetos realizan operaciones relacionadas con la información que poseen. Brinda beneficios como la conservación del encapsulamiento, el soporte de un bajo acoplamiento y una alta cohesión. (35)

En la solución que se propone, se puede apreciar la utilización de este patrón, en el vínculo que existe entre la vista (vistalogin), el controlador (controladorlogin) y el modelo (modellogin), los cuales realizan acciones en correspondencia a la información que manejan.

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos. Su propósito fundamental es encontrar un creador que debe ser conectado con el objeto producido en cualquier evento. El patrón Creador proporciona soporte a un bajo acoplamiento. (35)

Bajo acoplamiento: Estimula asignar una responsabilidad a una clase, de modo que su colaboración no incremente tanto el acoplamiento con otras clases, al nivel que produzca los resultados negativos propios

de un alto acoplamiento. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también reutilizables, que aumentan la oportunidad de una mayor productividad. (35)

En el diseño de la aplicación se hace uso de los patrones GRASP, especialmente del patrón Experto, Controlador, Alta Cohesión y Bajo Acoplamiento.

Patrones GOF

GOF es el acrónimo de Gans of Four (Pandilla de los cuatro), estos enmarcan los llamados patrones de diseño estructurales, creacionales y de comportamiento.

Patrones creacionales

Singleton

El objetivo de un patrón Singleton es el de tener una sola instancia de un objeto disponible para toda la aplicación que la contiene. Provee una instancia global de la siguiente forma:

1-Permitiendo que la clase Singleton cree una sola instancia de sí misma.

2-Permitiendo que otros objetos accedan esta única instancia a través de un método de la clase, la cual retorna una referencia de la única instancia. Este método es público y puede ser accedido globalmente.

3-Declarando el constructor de la clase Singleton como privado, para que otro objeto no pueda crear una instancia adicional de la clase. (34)

Ejemplo de la utilización del patrón en el framework CodeIgniter.

```
$CI =& get_instance();  
$CI->load->helper('url');  
$CI->config->item('base_url');
```

La función `get_instance()`; se pasa por referencia, lo que permite usar el objeto CodeIgniter original en lugar de hacer una copia de él.

Patrones de comportamiento

Observer (Observación)

El objetivo del patrón de Observación es la de desacoplar la información del objeto (atributos) del comportamiento (métodos), esto se logra creando un objeto que únicamente contiene información básica, y esta delega el comportamiento o las acciones a tomar a un objeto observador, el cual observa el objeto en cuestión, detecta o es notificado por los eventos que suceden en el mismo, y reacciona de una manera específica. (34)

Se evidencia el uso de la observación en el patrón arquitectónico MVC, ya que el objeto observado es el View, el observador es el Controller, y el objeto Model, contiene información sobre la cual el observador reacciona.

La utilización de la tecnología AJAX (JavaScript Asíncrono y XML) demuestra igualmente el uso de este patrón, puesto que el objeto observado que es el View, reacciona en dependencia de la respuesta que devuelva la llamada asíncrona.

2.8.3 Tarjetas CRC (Contenido – Responsabilidad - Colaboración)

Para el diseño de las aplicaciones, la metodología XP usa las tarjetas CRC. Éstas permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas CRC representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

Tabla 10 Tarjeta CRC

Nombre de la clase	
Responsabilidades	Colaboraciones

A continuación se muestran las tarjetas CRC generadas en la fase de diseño de la aplicación.

Tabla 11 Tarjeta CRC Módulo Autenticar.

Módulo Autenticar	
Responsabilidades	Colaboraciones
Autenticar.	controladorLogin. modelLogin. vistaLogin.

Tabla 12 Tarjeta CRC Módulo Administrador.

Módulo Administrador	
Responsabilidades	Colaboraciones
Gestionar cuentas en el sistema.	gestionarTrabajador.
Gestionar trabajador.	gestionarCuentas.
Realizar búsquedas filtradas.	controladorAsistencia.
Controlar asistencia diariamente.	modelGestionarTrabajador.
Gestionar salario.	modelAsistencia.
Gestionar evaluación de desempeño.	modelCuentas.
Exportar modelos.	vistaCuentas.
Exportar actas.	vistaTrabajador.
Gestionar capacitación.	vistaAsistencia
Gestionar estudios.	
Gestionar sanciones.	
Gestionar estimulación moral.	
Envío de correo electrónico a los trabajadores.	
Publicación de trabajos, informes y planillas.	

Tabla 13 Tarjeta CRC Módulo Usuario.

Módulo Usuario	
Responsabilidades	Colaboraciones
Realizar búsquedas filtradas.	controladorActas.
Exportar modelos.	controladorModelos.
Exportar actas.	controladorCorreo.

Envío de correo electrónico a los trabajadores. Publicación de trabajos, informes y planillas.	controladorArchivos. modelActas. modelModelos. modelArchivos. modelCorreo. vistaCorreo. vistaModelos vistaActas. vistaArchivos.
---	---

2.8.4 Diseño de la Base de Datos.

El diseño de la base de datos es algo que no se puede pasar por alto en la implementación, debido a que permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí. Ver modelo de datos (Anexo 2).

2.9 Conclusiones del capítulo

Durante el desarrollo del capítulo se expusieron los artefactos generados en las fases de exploración, planificación y diseño de la metodología XP. Asumiendo de esta forma una implementación por etapas correctamente descrita y detallada anteriormente. Durante esta fase aparecen factores primordiales para el éxito del producto como el diálogo entre clientes y desarrolladores los cuales exponen puntos clave en la decisión de si se va a continuar o no con el proyecto, argumentando entre lo deseable y lo posible.

Capítulo 3

Implementación y Pruebas

3.1 Introducción

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración, un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste. En el presente capítulo se detallan las tres iteraciones realizadas durante la etapa de construcción del sistema, exponiéndose las tareas generadas por cada historia de usuario, el diagrama de despliegue, así como las pruebas de aceptación efectuadas por el cliente.

3.2 Estándar de codificación

Los estándares de codificación son reglas que se utilizan para la escritura del código fuente y que incluye varios aspectos dentro del proceso de codificación. Éstos ayudan a crear un código más legible, reutilizable y de fácil mantenimiento.

3.2.1 Importancia de la utilización de estilos de código

La utilización de un buen estilo de código influye positivamente en las labores de mantenimiento de un software, como lo son la corrección de errores y el desarrollo de una nueva funcionalidad. Es realmente difícil que un programador mantenga una aplicación a lo largo de toda su vida; por tanto, el cambio de personal contribuye a un profundo estudio del código fuente. (36)

3.2.2 Elementos generales que conforman un estilo de codificación

Indentación

Permite ver la estructura jerárquica de la codificación sin tener que observar detalles del código, eso es posible sólo mirando la distribución de espacios en blanco y espacios escritos. En un fragmento de programa cuando se busca algo o se lee rápido, se puede visualizar la distribución de espacios, aunque

no se dedique tiempo a observar la estructura, el contenido o la concordancia de paréntesis, llaves, corchetes u otro separador. (36)

Notación

Permite definir un estándar de identificación a la hora de escribir los nombres de métodos, variables y clases. Los nombres deben darse con intencionalidad, independientemente de la distribución de símbolos o mayúsculas que se utilicen, por ejemplo cada nombre de variable debe transmitir su función en el código. (36)

Existen varias notaciones entre las cuales se encuentran la notación Camel, notación C, notación Húngara, entre otras. Para el desarrollo del sistema se utilizó la primera en su variante UpperCamelCase, la cual consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula.

Funciones y métodos

Clases: Los nombres comienzan con mayúscula.

Constantes: Se escriben en mayúscula en toda su extensión.

Funciones: Se escriben en minúscula separadas por un guión bajo.

Comentarios

Se utilizan comentarios del tipo `/* ... */`.

Llaves y líneas en blanco

Se usó una línea en blanco para separar agrupaciones lógicas del código.

Las llaves fueron utilizadas sobre líneas separadas y no sobre la misma línea. Además se usaron al mismo nivel del código que las contiene.

3.3 Fase de implementación

En esta fase, XP plantea la implementación de cada una de las historias de usuarios (HU). En un comienzo se hace un chequeo de cada una de las HU junto con el plan de iteraciones y se modifica en

caso de ser necesario. Como parte de este plan se crean tareas para ayudar a organizar la implementación exitosa de las HU. Estas tareas pueden escribirse en un lenguaje técnico, al contrario de las historias de usuario que son escritas en el lenguaje del cliente.

Por la planificación realizada con anterioridad se definieron 3 iteraciones de desarrollo del sistema.

A continuación se detallan cada una de las iteraciones.

3.3.1 Iteración 1

En esta iteración se implementan las historias de usuario de mayor prioridad, con el objetivo de obtener una versión con las funcionalidades más críticas del sistema para ser mostradas al cliente, y de esta manera tomar decisiones con respecto a los pasos a seguir para el desarrollo de las otras iteraciones.

Tabla 14 Historias de Usuario abordadas en la primera iteración.

Historia de Usuario	Tiempo de Implementación	
	Estimación	Real
Autenticar	1	1
Gestionar cuentas en el sistema.	1	1
Gestionar trabajador.	2	2
Realizar búsquedas filtradas.	1	1

A continuación se muestran las tareas efectuadas para cada una de las historias de usuario implementadas en esta iteración:

Tabla 15 Tarea 1 HU Autenticar

Tareas	
Número de la tarea: 1	Número de HU: 1
Nombre de la tarea: Creación de la tabla usuario en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 17/02/2012	Fecha fin: 17/02/2012
Programador responsable: Rajiv Rondón Padilla	
Descripción: Se crea en la base de datos la tabla usuario con sus diferentes campos.	

Tabla 16 Tarea 2 HU Autenticar

Tareas	
Número de la tarea: 2	Número de HU: 1
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 17/02/2012	Fecha fin: 17/02/2012
Programador responsable: Rajiv Rondón Padilla	
Descripción: La aplicación comprobará que exista el usuario dentro de la tabla usuarios, de ser verdadero permitirá iniciar sesión, sino mostrará un mensaje e invitará a autenticarse nuevamente.	

Tabla 17 Tarea 3 HU Gestionar cuentas.

Tareas	
Número de la tarea: 3	Número de HU: 2
Nombre de la tarea: Implementación de gestionar cuentas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha de inicio: 18/02/2012	Fecha fin: 18/02/2012
Programador responsable: Rajiv Rondón Padilla	
Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar cuentas.	

Tabla 18 Tarea 4 HU Gestionar trabajador

Tareas	
Número de la tarea: 4	Número de HU: 3
Nombre de la tarea: Creación de la tabla trabajadores en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 19/02/2012	Fecha fin: 19/02/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se crea en la base de datos la tabla correspondiente a trabajadores.	

Tabla 19 Tarea 5 HU Gestionar trabajador

Tareas	
Número de la tarea: 5	Número de HU: 3
Nombre de la tarea: Implementación de gestionar trabajador	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.3
Fecha de inicio: 20/02/2012	Fecha fin: 24/02/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar trabajadores.	

Tabla 20 Tarea 6 HU Realizar búsquedas filtradas

Tareas	
Número de la tarea: 6	Número de HU: 4
Nombre de la tarea: Realizar búsquedas filtradas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 25/02/2012	Fecha fin: 27/02/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: El sistema permitirá la opción de realizar las diferentes búsquedas necesarias.	

3.3.2 Iteración 2

En esta iteración se implementan las historias de usuario con un nivel de complejidad alta y media y se continúan perfeccionando las historias de usuario implementadas en la iteración anterior.

Tabla 21 Historias de Usuario abordadas en la segunda iteración.

Historia de Usuario	Tiempo de Implementación	
	Estimación	Real
Controlar asistencia diariamente.	1	1
Gestionar Salario.	1	1
Reportar evaluación de desempeño.	1	1
Exportar modelos.	2	2

Tabla 22 Tarea 7 HU Controlar asistencia diariamente.

Tareas	
Número de la tarea: 7	Número de HU: 5
Nombre de la tarea: Creación de la tabla asistencia en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 3/03/2012	Fecha fin: 8/03/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se crea en la base de datos la tabla asistencia.	

Tabla 23 Tarea 8 HU Controlar asistencia diariamente.

Tareas	
Número de la tarea: 8	Número de HU: 5
Nombre de la tarea: Controlar asistencia diariamente.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 9/03/2012	Fecha fin: 19/03/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: El sistema dará la opción de controlar la asistencia diariamente.	

Tabla 24 Tarea 9 HU Gestionar salario.

Tareas	
Número de la tarea: 9	Número de HU: 6
Nombre de la tarea: Creación de la tabla salario en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.3
Fecha de inicio: 20/03/2012	Fecha fin: 20/03/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se crea en la base de datos la tabla correspondiente a salario.	

Tabla 25 Tarea 10 HU Gestionar salario.

Tareas	
Número de la tarea: 10	Número de HU: 6
Nombre de la tarea: Gestionar salario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 22/04/2012	Fecha fin: 23/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar salario.	

Tabla 26 Tarea 11 HU Reportar evaluación de desempeño.

Tareas	
Número de la tarea: 11	Número de HU: 7
Nombre de la tarea: Reportar evaluación de desempeño.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha de inicio: 20/03/2012	Fecha fin: 22/03/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Permite exportar la evaluación de desempeño.	

Tabla 27 Tarea 12 HU Exportar modelos.

Tareas	
Número de la tarea: 12	Número de HU: 8
Nombre de la tarea: Exportar modelos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 22/03/2012	Fecha fin: 29/03/2012
Programador responsable: Rajiv Rondón Padilla	
Descripción: Permite exportar modelo de evaluación de desempeño, certificado médico, certificado de vinculación para docentes, asistencia y certificado de evaluación de desempeño.	

3.3.3 Iteración 3

En esta iteración se implementan las historias de usuario con una complejidad baja, se terminan de perfeccionar los módulos implementados en las iteraciones anteriores y se entrega la aplicación al cliente.

Tabla 28 Historias de Usuario abordadas en la segunda iteración.

Historia de Usuario	Tiempo de Implementación	
	Estimación	Real
Exportar actas.	1	1
Gestionar capacitación.	1	1
Gestionar estudios.	1	1
Gestionar sanciones.	1	1
Gestionar estimulación moral.	1	1
Envío de correo electrónico a los trabajadores.	1	1
Publicación de trabajos, informes y planillas.	1	1

Tabla 29 Tarea 13 HU Exportar actas.

Tareas	
Número de la tarea: 13	Número de HU: 9
Nombre de la tarea: Exportar actas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 1/04/2012	Fecha fin: 5/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: El sistema permitirá exportar actas de admisión, advertencia, comité de ingreso, amonestación y boleta de admisión.	

Tabla 30 Tarea 14 HU Gestionar capacitación.

Tareas	
Número de la tarea: 14	Número de HU: 10
Nombre de la tarea: Creación de la tabla capacitación en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2

Fecha de inicio: 6/04/2012	Fecha fin: 6/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se crea en la base de datos la tabla correspondiente a capacitación.	

Tabla 31 Tarea 15 HU Gestionar capacitación.

Tareas	
Número de la tarea: 15	Número de HU: 10
Nombre de la tarea: Gestionar capacitación.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 7/04/2012	Fecha fin: 8/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar una capacitación.	

Tabla 32 Tarea 16 HU Gestionar estudios.

Tareas	
Número de la tarea: 16	Número de HU: 11
Nombre de la tarea: Creación de la tabla educación superior en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 9/04/2012	Fecha fin: 9/04/2012
Programador responsable: Rajiv Rondón Padilla	
Descripción: Se crea en la base de datos la tabla correspondiente a educación superior.	

Tabla 33 Tarea 17 HU Gestionar estudios.

Tareas	
Número de la tarea: 17	Número de HU: 11
Nombre de la tarea: Gestionar estudios.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.3
Fecha de inicio: 10/04/2012	Fecha fin: 12/04/2012
Programador responsable: Rajiv Rondón Padilla.	

Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar un trabajador en estudios superiores.

Tabla 34 Tarea 18 HU gestionar sanciones.

Tareas	
Número de la tarea: 18	Número de HU: 12
Nombre de la tarea: Creación de la tabla sanciones en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 13/04/2012	Fecha fin: 13/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se crea en la base de datos la tabla correspondiente a sanciones.	

Tabla 35 Tarea 19 HU gestionar sanciones.

Tareas	
Número de la tarea: 19	Número de HU: 12
Nombre de la tarea: Gestionar sanciones.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 15/04/2012	Fecha fin: 17/04/2012
Programador responsable: Rajiv Rondón Padilla	
Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar sanciones.	

Tabla 36 Tarea 20 HU Gestionar estimulación moral.

Tareas	
Número de la tarea: 20	Número de HU: 13
Nombre de la tarea: Creación de la tabla estimulación moral en la base de datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 18/04/2012	Fecha fin: 18/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se crea en la base de datos la tabla correspondiente a estimulación moral.	

Tabla 37 Tarea 21 HU Gestionar estimulación moral.

Tareas	
Número de la tarea: 21	Número de HU: 13
Nombre de la tarea: Gestionar estimulación moral.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 19/04/2012	Fecha fin: 21/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se implementan las funcionalidades que permiten mostrar, adicionar, modificar y eliminar un trabajador con estimulación moral.	

Tabla 38 Tarea 22 HU Envío de correo electrónico a los trabajadores.

Tareas	
Número de la tarea: 22	Número de HU: 14
Nombre de la tarea: Envío de correo electrónico a los trabajadores.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha de inicio: 21/04/2012	Fecha fin: 22/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se implementa la funcionalidad de enviar correo electrónico.	

Tabla 39 Tarea 23 HU Publicación de trabajos, informes y planillas.

Tareas	
Número de la tarea: 23	Número de HU: 15
Nombre de la tarea: Publicación de trabajos, informes y planillas.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 23/04/2012	Fecha fin: 23/04/2012
Programador responsable: Rajiv Rondón Padilla.	
Descripción: Se implementa la funcionalidad de publicar trabajos, informes y planillas.	

3.4 Diagrama de Despliegue

El diagrama de despliegue muestra las relaciones físicas entre componentes de hardware y software en el sistema. En este caso la aplicación se encuentra hosteada en un servidor web apache y maneja los datos persistentes sobre un sistema gestor de base de datos MySQL. Los dos servidores están alojados en la misma máquina, ya que el Departamento de Producción Audiovisual (DPA) así lo designó.

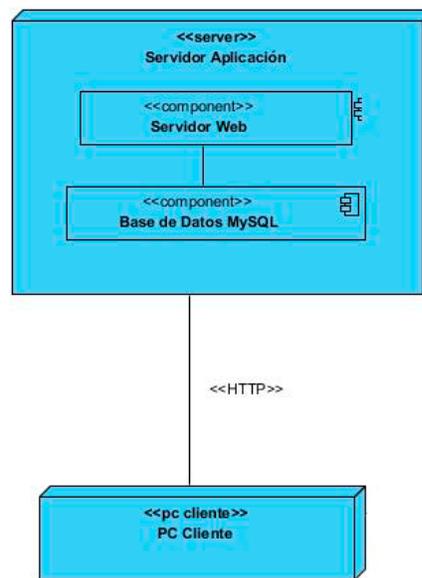


Figura 1 Diagrama de Despliegue

3.5 Pruebas

Uno de los pilares fundamentales de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se han implementado. 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. La metodología XP divide las pruebas en dos grupos:

Pruebas unitarias: Desarrolladas por los programadores y encargadas de verificar el código de forma automática.

Pruebas de aceptación: Destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

3.5.1 Pruebas de aceptación.

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado, éstas posibilitan evaluar las distintas tareas en las que ha sido dividida una historia de usuario y asegurar su correcto funcionamiento, una HU no se considera terminada hasta que no ha pasado sus pruebas de aceptación. Ver pruebas de aceptación (Anexo 3).

Se realizaron tres iteraciones, detectándose varias no conformidades en la primera y segunda iteración, las mismas fueron solucionadas en la segunda y tercera iteración. Para tener una visión más detallada ver (Anexo 4).

3.6 Conclusiones del capítulo

En este capítulo se hizo alusión a las fases de implementación y pruebas. Se desarrollaron las tareas correspondientes para dar solución a las historias de usuario y las pruebas de aceptación que propician al cliente seguridad y conformidad ante el sistema. Con el fin de este capítulo se da por terminada la propuesta que trae este trabajo.

Conclusiones Generales:

Como resultado de este trabajo se realizó un estudio de los sistemas de gestión de información de Recursos Humanos existentes en el ámbito foráneo y nacional. Además de las tecnologías y herramientas disponibles para el desarrollo de la aplicación, seleccionando las más adecuadas según diversos criterios. A partir de los requisitos identificados, según la metodología definida y usando UML como lenguaje de modelado, se realizó la planificación y diseño, permitiendo abarcar todas las funcionalidades y capacidades con las que el sistema debía cumplir.

La implementación de la aplicación se realizó siguiendo el plan de iteraciones trazado, el cumplimiento estricto del mismo permitió obtener un sistema cumpliendo con el cronograma inicial de la presente investigación.

La estrategia de pruebas seguida posibilitó obtener un producto probado desde diferentes puntos de vista, donde las pruebas realizadas arrojaron resultados positivos. Por todo lo anteriormente planteado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente.

Recomendaciones:

Como resultado del proceso de investigación y desarrollo del sistema han surgido algunas ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento de la aplicación, las cuales aportarían una mejor experiencia a los usuarios finales. A continuación se listan las mismas.

1. Continuar agregándole funcionalidades al sistema, de acuerdo a nuevas expectativas de los usuarios.
2. Automatizar otros procesos de la Dirección de Producción Audiovisual.

Referencias Bibliográficas

1. **PRENDES, Lourdes M.** *Caracterización de un Sistema de Gestión de Información Científico Tecnológica con enfoque a procesos: garantía para la mejora continua.* Ciudad de la Habana, 2006.
2. **PIEDRA, Armando Jesús.** *Implementación del Sistema Integrado de Gestión de Capital Humano (SIGCH) en las empresas cubanas contemporáneas,* 2009.
3. **MICROSOFT. Microsoft.com.** [En línea] 2009. [Citado el: 9 de 1 de 2012.] Disponible en Web: <http://www.microsoft.com/dynamics/es/xl/products.aspx>.
4. **HUMSOFTWARE. Humsoftware.com.** [En línea]. [Citado el: 9 de 1 de 2012.] Disponible en Web: <http://www.humsoftware.com/Paginas/Sel-Index.htm>.
5. **PROGRAMAS. programas.com.** [En línea] 2011. [Citado el: 9 de 1 de 2012.] Disponible en Web: http://www.programas.com/descargar_/gesrrhh.
6. **PLUSFORMACION. Plusformacion.com.** [En línea] [Citado el: 9 de 1 de 2012.] Disponible en Web: http://www.plusformacion.com/Recursos/r/Gestion-por-Procesos-Recursos-Humanos?quicktabs_ofertas_relacionadas_quicktab=2#softrh.
7. **RHERNANDO. Rhernando.net.** [En línea] [Citado el: 28 de 2 de 2012.] Disponible en Web: http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
8. **MOLPECERES, Alberto.** *Procesos de desarrollo: RUP, XP y FDD.* AT javahispano DOT.org, 2002.
9. **AMARO CALDERON, Jorge Carlos.** [En línea] 2007. [Citado el: 10 de 2 de 2012.] Disponible en Web: <http://seccperu.org/files/Metodologias%20Agiles.pdf>.
10. **PROYECTOAGILES. Proyectoagiles.** *Qué es SCRUM?* [En línea] [Citado el: 2011 de 11 de 10.] Disponible en Web: www.proyectosagiles.org.
11. **KNIBERG, Henrik.** *SCRUM Y XP DESDE LAS TRINCHERAS.* Estados Unidos de América, 2007.

12. **SOFTENG. Softeng.** [En línea] 2010. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum>.
13. **UML. Uml.** [En línea] 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.uml.org>.
14. **VISUALPARADIGM. VisualParadigm.** [En línea] 2012. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.visual-paradigm.com>.
15. **RATIONAL. Rational.** [En línea] 31 de 3 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.rational.com.ar/herramientas/roseenterprise.html>.
16. **MITECNOLOGICO. Mitecnológico.** [En línea] 13 de 5 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.
17. **EGUILYZ PEREZ, Javier.** *Introducción a JavaScript, 2009.* [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.librosweb.es/javascript>.
18. **LIBROSWEB. LibrosWeb.** [En línea] [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.librosweb.es/xhtml>.
19. **EGUILYZ PEREZ, Javier.** *Introducción a css, 2008.* [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.librosweb.es/css>.
20. **ZELDMAN, Jeffrey.** *Diseño con estándares Web.* Ediciones ANAYA MULTIMEDIA, 2004.
21. **GALLEGO VAZQUEZ, José Antonio.** *Desarrollo Web con PHP y MySQL.* Ediciones ANAYA MULTIMEDIA, 2006.
22. **ELCODIGOK. ElCodigok.** [En línea] 14 de 5 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.elcodigok.com.ar/2010/09/caracteristicas-de-un-excelente-entorno-de-desarrollo-integrado>.
23. **BITSBETA. BitsBeta.** [En línea] 5 de 12 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://bitsbeta.com/netbeans-ide-grafica-programacion>.

24. **MONTERO GARRIDO, Jesús Manuel.** *Plataforma Eclipse*, 2004.
25. **POTENCIER, Fabien.** *Symfony la guía definitiva*. [Citado el: 10 de 1 de 2012.] Disponible en Web: http://www.librosweb.es/symfony_1_0, 2008.
26. **LIBROJQUERY.** *LibroJquery*. [En línea] 2012. [Cited: 2 10, 2012.] Disponible en Web: <http://librojquery.com>.
27. **MAESTROSDDELWEB.** *MaestrosdelWeb*. [En línea] 2012. [Cited: 2 10, 2012.] Disponible en Web: <http://www.maestrosdelweb.com/editorial/javascript-facil-y-rapido-con-jquery>.
28. **FREDERICK, Shea.** *Learning Ext*. Akshara Aware, 2008.
29. **CONOCIMIENTO VIRTUAL, Colombia.** *Code Igniter Spanish User Guide*. [En línea] 2012. [Cited: 2 10, 2012.] Disponible en Web: <http://www.conocimientovirtual.edu.co>.
30. **MATEU, Carles.** *Desarrollo de aplicaciones web*. Fundació per a la Universitat Oberta de Catalunya (Formación de Posgrado), 2004.
31. **MICROSOFT.** *Microsoft*. [En línea] 2012. [Cited: 1 10, 2012.] Disponible en Web: <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.
32. **GALLEGO, Jose Antonio.** *Desarrollo Web con PHP y My SQL*. ANAYA Multimedia, 2006.
33. **MYSQL, MySql.** [En línea] 2012. [Cited: 1 10, 2012.] Disponible en Web: <http://www.mysql.com/why-mysql>.
34. **PARRA, Jose David.** *Guía de Patrones, Prácticas y Arquitectura .NET*, 2012.
35. **LARMAN, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, 1999.
36. **CALLEJA, M.A.** *Estándares de codificación*, 2000.

Bibliografía

1. **PRENDES, Lourdes M.** *Caracterización de un Sistema de Gestión de Información Científico Tecnológica con enfoque a procesos: garantía para la mejora continua.* Ciudad de la Habana, 2006.
2. **PIEDRA, Armando Jesús.** *Implementación del Sistema Integrado de Gestión de Capital Humano (SIGCH) en las empresas cubanas contemporáneas,* 2009.
3. **MICROSOFT. Microsoft.com.** [En línea] 2009. [Citado el: 9 de 1 de 2012.] Disponible en Web: <http://www.microsoft.com/dynamics/es/xl/products.aspx>.
4. **HUMSOFTWARE. Humsoftware.com.** [En línea]. [Citado el: 9 de 1 de 2012.] Disponible en Web: <http://www.humsoftware.com/Paginas/Sel-Index.htm>.
5. **PROGRAMAS. programas.com.** [En línea] 2011. [Citado el: 9 de 1 de 2012.] Disponible en Web: http://www.programas.com/descargar_/gesrrhh.
6. **PLUSFORMACION. Plusformacion.com.** [En línea] [Citado el: 9 de 1 de 2012.] Disponible en Web: http://www.plusformacion.com/Recursos/r/Gestion-por-Procesos-Recursos-Humanos?quicktabs_ofertas_relacionadas_quicktab=2#softrh.
7. **RHERNANDO. Rhernando.net.** [En línea] [Citado el: 28 de 2 de 2012.] Disponible en Web: http://www.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
8. **MOLPECERES, Alberto.** *Procesos de desarrollo: RUP, XP y FDD.* AT javahispano DOT.org, 2002.
9. **AMARO CALDERON, Jorge Carlos.** [En línea] 2007. [Citado el: 10 de 2 de 2012.] Disponible en Web: <http://seccperu.org/files/Metodologias%20Agiles.pdf>.
10. **PROYECTOAGILES. Proyectoagiles.** *Qué es SCRUM?* [En línea] [Citado el: 2011 de 11 de 10.] Disponible en Web: www.proyectosagiles.org.
11. **KNIBERG, Henrik.** *SCRUM Y XP DESDE LAS TRINCHERAS.* Estados Unidos de América, 2007.

12. **SOFTENG. Softeng.** [En línea] 2010. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum>.
13. **UML. Uml.** [En línea] 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.uml.org>.
14. **VISUALPARADIGM. VisualParadigm.** [En línea] 2012. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.visual-paradigm.com>.
15. **RATIONAL. Rational.** [En línea] 31 de 3 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.rational.com.ar/herramientas/roseenterprise.html>.
16. **MITECNOLOGICO. Mitecnológico.** [En línea] 13 de 5 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.
17. **EGUILYZ PEREZ, Javier.** *Introducción a JavaScript*, 2009. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.librosweb.es/javascript>.
18. **LIBROSWEB. LibrosWeb.** [En línea] [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.librosweb.es/xhtml>.
19. **EGUILYZ PEREZ, Javier.** *Introducción a css*, 2008. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.librosweb.es/css>.
20. **ZELDMAN, Jeffrey.** *Diseño con estándares Web*. Ediciones ANAYA MULTIMEDIA, 2004.
21. **GALLEGO VAZQUEZ, José Antonio.** *Desarrollo Web con PHP y MySQL*. Ediciones ANAYA MULTIMEDIA, 2006.
22. **ELCODIGOK. ElCodigok.** [En línea] 14 de 5 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://www.elcodigok.com.ar/2010/09/caracteristicas-de-un-excelente-entorno-de-desarrollo-integrado>.
23. **BITSBETA. BitsBeta.** [En línea] 5 de 12 de 2011. [Citado el: 10 de 1 de 2012.] Disponible en Web: <http://bitsbeta.com/netbeans-ide-grafica-programacion>.

24. **MONTERO GARRIDO, Jesús Manuel.** *Plataforma Eclipse*, 2004.
25. **POTENCIER, Fabien.** *Symfony la guía definitiva*. [Citado el: 10 de 1 de 2012.] Disponible en Web: http://www.librosweb.es/symfony_1_0, 2008.
26. **LIBROJQUERY.** *LibroJquery*. [En línea] 2012. [Cited: 2 10, 2012.] Disponible en Web: <http://librojquery.com>.
27. **MAESTROSDDELWEB.** *MaestrosdelWeb*. [En línea] 2012. [Cited: 2 10, 2012.] Disponible en Web: <http://www.maestrosdelweb.com/editorial/javascript-facil-y-rapido-con-jquery>.
28. **FREDERICK, Shea.** *Learning Ext. Akshara Aware*, 2008.
29. **CONOCIMIENTO VIRTUAL, Colombia.** *Code Igniter Spanish User Guide*. [En línea] 2012. [Cited: 2 10, 2012.] Disponible en Web: <http://www.conocimientovirtual.edu.co>.
30. **MATEU, Carles.** *Desarrollo de aplicaciones web*. Fundació per a la Universitat Oberta de Catalunya (Formación de Posgrado), 2004.
31. **MICROSOFT.** *Microsoft*. [En línea] 2012. [Cited: 1 10, 2012.] Disponible en Web: <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.
32. **GALLEGO, José Antonio.** *Desarrollo Web con PHP y My SQL*. ANAYA Multimedia, 2006.
33. **MYSQL, MySql.** [En línea] 2012. [Cited: 1 10, 2012.] Disponible en Web: <http://www.mysql.com/why-mysql>.
34. **PARRA, José David.** *Guía de Patrones, Prácticas y Arquitectura .NET*, 2012.
35. **LARMAN, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, 1999.
36. **CALLEJA, M.A.** *Estándares de codificación*, 2000.
37. **ALVAREZ, Sara.** [En línea] 7 31, 2007. [Cited: 1 9, 2012.] Disponible en Web: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

38. **BRADENBAUGH, Jerry.** *Programación de aplicaciones web con javaScrip.* España : EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2000.
39. **SOLIS CALERO, Manuel.** *Una explicación de la programación extrema (XP),* 2003.
40. **BECK, Kent.** *Extreme Programming Explained,* 1999.
41. **CANOS, Jose H.** *Métodologías Ágiles en el Desarrollo de Software.* Universidad Politécnica de Valencia.
42. **VEGA BRICENO, Edgar Armando.** *GestioPolis.* [En línea] junio 2006.

Glosario de Términos:

AJAX: Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

DOM: Document Object Model. Modelo de Objetos de Documento.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

HTML: Hypertext Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

PHP: Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

Software: Programas de sistema, utilerías o aplicaciones expresadas en un lenguaje de máquina.

SQL: Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.

XML: Extensible Markup Language. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.