



**Universidad de Matanzas
Facultad de Ciencias Técnicas
Especialidad Ingeniería Informática**



**Trabajo para optar por el Título de Ingeniero en
Informática.**

Tema: Aplicación web para la informatización de la gestión de expedientes laborales en el Departamento de Recursos Humanos

Autor: Noel Pérez Suria

Tutor: Jorge Ilich Noda

Matanzas, 2018

**“...Toda la gloria del mundo cabe en un grano de
maíz...”**

José Martí

Dedicado :

A mis padres, que me impulsaron a seguir adelante cuando el viento soplaba fuerte y las nubes eran grises.

A mis amigos, aquellos que extendieron una mano de ayuda en el momento en que más fue necesaria.

Agradecimientos:

A todos los que estuvieron allí en estos años en la universidad, a aquellos con los que compartí dificultades y creé grandes recuerdos.

Declaración de autoría

Declaro que yo, **Noel Pérez Suria** me declaro único autor de este documento, autorizando a la Universidad de Matanzas, especialmente al Departamento de Informática que haga uso del mismo como entienda pertinente.

Para que así conste, firmo a los _____, 2018.

Autor

Raúl Suárez Hernández

Tutor

Jorge Ilich Noda.

Opinión del autor del Trabajo de Diploma.

Título: Aplicación web para la informatización de la gestión de expedientes laborales en el Departamento de Recursos Humanos

Autor: Noel Pérez Suria

Jorge Ilich Noda

Resumen:

En los tiempos actuales, donde las masivas cantidades de información circundantes han dificultado el trabajo de aquellas personas encargadas de manejar documentos, se ha convertido en una necesidad la elaboración de sistemas que permitan informatizar dichos datos, facilitando su posterior uso y almacenamiento. En esta investigación se hace un estudio del proceso de gestión de los expedientes laborales en el Departamento de Recursos Humanos de la Universidad de Matanzas, Camilo Cienfuegos, en cuyo lugar se refleja la necesidad de contar con un mecanismo factible que mejore el desempeño en el trabajo llevado a cabo por los técnicos. La opción que se propone para ello es la elaboración de una aplicación web que sea capaz de gestionar el proceso de control de los expedientes laborales en el departamento. Para su elaboración se pretende utilizar el framework de desarrollo Symfony a través de los lenguajes de programación php, html, css y java script. Se empleará Wamp como servidor web y My SQL como sistema gestor de base de datos. Para completar esta tarea se utilizará la metodología ágil Programación Extrema (XP).

Summary:

On the current times, where the massive amount of information have increased the difficulty of the work of those people in charge with handling documents, it has become a necessity the elaboration of systems that allows to digitalize those data, making easier its later use and storing. In this research we make a study of the process of management of the expedients in the Human Resource Department on the University Camilo Cienfuegos, a place that reflects the need of counting with a proper mechanism that improves the performance in the work carried on by the technicians. The option we propose is the development of a web application capable of managing the control process of the expedients in the department. For its making we pretend to use the development framework Symfony, implementing it with the programming languages php, html, css y java script. We will employ Wamp as web server and My SQL as database management system. To complete this task we will use the agile methodology Extreme Programming (XP).

Índice

Introducción.....	
Capítulo I: Marco Teórico Referencial.....	
1.1 Introducción.....	
1.2 Fundamentación teórica.....	
1.3 Objeto de Estudio.....	
1.4 Antecedentes del Trabajo.....	
1.5 Métodos de Investigación.....	
1.6 Tendencias Tecnológicas.....	
1.6.1 Fundamentación de la metodología utilizada.....	
1.6.2 XP- eXtremeProgramming.....	
1.6.3- Aplicación web.....	
1.6.4 Patrón de diseño Modelo Vista Controlador (MVC)...	
1.6.5 Gestor de Base de Datos.....	
1.6.6 Framework Bootstrap.....	
1.6.7 Framework Symfony 2.....	
1.6.8 Lenguajes de Programación.....	
Capítulo II: Solución teórica al problema científico.....	
2.1 Introducción.....	
2.2 Descripción de la solución propuesta.....	
2.3 Requisitos funcionales.....	
2.4 Requisitos no funcionales.....	

2.5	Integrantes del equipo.....	
2.6	Diagrama de procesos.....	
2.7	Historias de usuario.....	
2.7.1	Resumen de Historias de usuarios.....	
2.8	Estimación por puntos de función.....	
2.9	Planificación de iteraciones.....	
2.10	Plan de entrega.....	
2.11	Tareas de iteración.....	
2.12	Diseño de Tarjetas CRC.....	
2.13	Análisis de los Costos(Modelo matemático COCOMO II).....	
Capítulo III: Resultados del Trabajo Desarrollado. Elementos de la Validación Práctica de la Propuesta de Solución del Problema Científico.....		
3.1	Introducción.....	
3.2	Pruebas.....	
3.2.1	Plan de Pruebas.....	
3.2.2	Pruebas de Aceptación.....	
3.2.3	Pruebas Funcionales.....	
3.3	Análisis de los resultados obtenidos.....	
Conclusiones.....		
Recomendaciones.....		
Bibliografía.....		

Introducción:

En la Universidad de Matanzas Camilo Cienfuegos en el departamento de Recursos Humanos específicamente en el departamento de Procesos Laborales se gestionan datos importantes de los trabajadores en la Universidad. Actualmente en el departamento general de recursos Humanos se cuenta con el Software ASSETS, pero el departamento de Procesos Laborales no lo utiliza debido a que el mismo presenta un nivel de complejidad demasiado alto, demandando de sus usuarios varios conocimientos que necesariamente no tienen por qué tener, además de que no les brinda todas las opciones que realmente necesita el departamento para la realización de su trabajo. Nuestra propuesta es desarrollar una aplicación web que resuelva todas estas dificultades para cuando se requiera de la información, que esta sea manejable y segura. Con este trabajo proponemos una solución simple y fácil de usar para el personal de Recursos Humanos en el departamento de Procesos Laborales, ganando así un poco de experiencia para proyectos futuros.

Luego de un estudio detallado de la situación anteriormente reflejada se logró identificar el siguiente **problema científico**: ¿Cómo realizar de manera simple, pero adecuada la gestión de los procesos para el trabajo con los expedientes laborales de los trabajadores en el departamento de Recursos Humanos?

Hipótesis:

Si se desarrolla el sistema web para gestionar el registro y control de procesos para los expedientes laborales de los trabajadores, entonces se logrará evitar las equivocaciones, pérdidas de datos, duplicación de información en los procesos de registro y posibilitará un buen control del proceso de contratación y evaluación, además se contribuirá a mejorar la eficiencia del trabajo en el departamento de Recursos Humanos de la Universidad.

El **objetivo general** que se persigue con este trabajo es desarrollar una aplicación web para informatizar los procesos de gestión de los expedientes

laborales para el Departamento de Recursos Humanos de la Universidad de Matanzas

Definiéndose como **objeto de estudio** los procesos de gestión de los expedientes laborales, y como **campo de acción** la informatización de los procesos de gestión de los expedientes laborales para el Departamento de Recursos Humanos de la Universidad de Matanzas.

Para darle cumplimiento al objetivo general se trazarán los siguientes **objetivos específicos:**

1. Elaborar el marco teórico referencial del proceso de informatización del Departamento de Recursos Humanos
2. Implementar una aplicación web para el Departamento de Recursos Humanos de la Universidad
3. Validar la solución obtenida a través de pruebas de software
4. Implantar el software en el Departamento de Recursos Humanos de la Universidad

Los **Métodos empleados** para la elaboración del software fueron:

Métodos Empíricos:

Observación

Entrevistas y encuestas

Análisis de documentos

Métodos Teóricos:

Método de análisis histórico-lógico

Método de análisis y síntesis

Método inductivo - deductivo

Técnicas para la captura de requisitos utilizadas:

Dentro de las técnicas más utilizadas para la captura de requerimientos están las entrevistas realizadas a los usuarios, la tormenta de ideas y el análisis de sistemas ya automatizados que estén vinculados con el producto que se desea obtener.

Entrevistas: A través de esta técnica nos acercamos al problema de una forma natural y se obtiene una amplia visión del trabajo a realizar y las necesidades del usuario.

Tormenta de ideas: Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas.

Sistemas existentes: Consiste en estudiar aplicaciones desarrolladas que estén relacionadas con el sistema que se desarrollará.

Resumen por capítulos

En consecuencia a todo lo planteado hasta ahora el documento queda estructurado de la siguiente manera:

Capítulo I: Marco teórico referencial

Fundamentación teórica y tendencias tecnológicas. Se abordan los antecedentes del trabajo, los conceptos fundamentales asociados al problema y se describen las tendencias tecnológicas actuales que se seleccionaron para el desarrollo del Sistema.

Capítulo II: Descripción de la solución propuesta

Se efectúa la descripción de la situación propuesta, una planificación inicial del proyecto con el empleo de la metodología XP (Extreme Programming).

Capítulo III: Validación de la solución

Construcción de la solución propuesta y el análisis de los resultados obtenidos. En este capítulo se construye a solución. Se efectúa un análisis de los resultados obtenidos basándose en la opinión del cliente. Se realizan pruebas al sistema. Por último aparecen los acápites de conclusiones, recomendaciones y anexos.

Capítulo I: Marco Teórico-Referencial

1.1 Introducción

En el actual capítulo se estudian los conceptos fundamentales que están relacionados a la fundamentación del dominio del problema. Se discutirán las metodologías, las tecnologías, los antecedentes relacionados al trabajo y las herramientas empleadas para la implementación de una aplicación web.

1.2 Fundamentación Teórica

La gestión de la información es algo fundamental en cualquier entidad, especialmente cuando se refiere al control sobre la información personal de los trabajadores de un lugar determinado. La aplicación web que proponemos permitirá manejar de manera sencilla los datos personales de los trabajadores, así como los procesos de gestión de sus expedientes laborales, permitiendo agilizar el trabajo y acceder de manera simple a los misma en caso de ser necesario.

1.3 Objeto de estudio

Esta enfocado en el proceso de realización de la gestión de los expedientes laborales, así como el control de los datos de profesores y trabajadores y cualquier cambio en el que estos deban de incurrir en el transcurso del tiempo.

1.4 Antecedentes

El Departamento de Recursos Humanos de la Universidad de Matanzas contaba originalmente con el software ASSETS NS, un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos, pero el mismo comenzo a demostrar ser ineficiente en la obtención de los resultados esperados por quienes lo utilizaban en el departamento. ASSETS es un software potente y altamente utilizado, pero cuenta con la desventaja de ser inusualmente complejo y para explotar por completo sus ventajas es necesario contar con una serie de conocimientos que no toda persona posee. Tras llevar a cabo varias entrevistas en el departamento y comprobar que su utilización era demasiado limitada en manos de los trabajadores del local se decidió apostar por la elaboración de un software más simple, pero que permitiera a usuarios sin amplios conocimientos informáticos utilizarlo, para llevar a cabo todo el trabajo relacionado con la gestión de los expedientes laborales.

1.5 Métodos de investigación empleados

Métodos teóricos empleados:

Método de análisis histórico-lógico: Permitió analizar la trayectoria del trabajo realizado en el departamento, concerniente al manejo de los expedientes laborales, así como de los cambios llevados a cabo en el salario de los trabajadores durante las operaciones.

Análisis y síntesis: este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división mental del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.

Inducción-deducción: su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

Métodos empíricos empleados:

Para poner en práctica estos métodos se llevan a cabo las siguientes técnicas:

Observación científica: Acompaña el desarrollo de la investigación desde sus orígenes, a través de la cual se pudo conocer el estado de los procesos que se llevaban a cabo en la entidad.

Entrevista: Al ser el entrevistado el cliente que propuso el desarrollo del proyecto resultó extremadamente útil, sobre todo para la captura de requisitos previos a la implementación de la interfaz.

Análisis de documentos: Se estudiaron varios de los documentos existentes en los que se reflejaban el trabajo que se realizaba en la entidad.

1.6 Tendencias Tecnológicas

1.6.1 Fundamentación de la metodología utilizada

Las metodologías de desarrollo de software son un conjunto de pasos y procedimientos que deben seguirse para llevar a cabo el desarrollo de software con calidad, éstas brindan un conjunto de detalles organizativos, añadiendo rigurosidad y normas, permitiendo que los integrantes de un equipo de desarrollo puedan seguir un criterio común a la hora de realizar las tareas durante el desarrollo de un software.(Balaguera 2015)

Actualmente existen dos grandes grupos de metodologías de desarrollo de software:

Metodologías ágiles: dan mayor valor a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Se basa en la filosofía de que es más importante desarrollar software que funcione, que conseguir una buena documentación y es más importante responder ante un cambio, que seguir estrictamente un plan.(Letelier 2010)

Metodologías tradicionales: llevan un control estricto del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto.

Es más factible la utilización de una metodología ágil ya que presentan diversas ventajas, entre las que podemos destacar:

El cliente es parte del equipo de desarrollo.

La arquitectura se va definiendo y mejorando a lo largo del proyecto.

Especialmente preparadas para cambios durante el proyecto.

1.6.2 XP- eXtremeProgramming

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo esto trae consigo una mejor fluidez entre los integrantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

XP tiene como característica la comunicación directa y amplia con el cliente y entre los miembros del equipo de desarrollo, buena respuesta ante los cambios sin importar la etapa del ciclo de vida de la aplicación donde se encuentre, se promueve la práctica de lo que brinde valor real al software antes que la documentación exhaustiva, la simplicidad en las soluciones y, generalmente los proyectos desarrollados bajo esta metodología tienen un ciclo de vida corto..(Sánchez, Letelier et al. 2003), (Campos and Martínez 2015), (Tracz 2002).

Por lo tanto se pondrá en práctica dicha metodología.

1.6.3- Aplicación web

Una aplicación web es cualquier aplicación que es accedida vía web por una red como internet o una intranet. En general, el término también se utiliza para

designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación.(Bello 2013)

1.6.4 Patrón de diseño Modelo Vista Controlador (MVC):

El patrón MVC es una arquitectura de diseño software para separar los componentes de aplicación en tres niveles, interfaz de usuario, lógica de control y lógica de negocio (González and Romero 2012),(Bascón 2010),(Espinosa, Sagredo et al. 2012).

Modelo: Es la capa encargada de encapsular toda la lógica de negocio de nuestra aplicación. Esta capa se puede subdividir en varias:

Lógica de negocio: Contiene clases o varias para constituir lo referente a la capa de aplicación, se encarga de atender a las peticiones de los controladores y así dar una respuesta acorde con lo recibido.

Capa de datos: Se encarga de gestionar toda la interconexión con el SGBD, así mismo, puede contener un gestor ORM para su aprovechamiento máximo y mejor mantenimiento. Solo se comunica con la lógica de negocio.

Helpers: Llamados “ayudantes” apoyan tanto al controlador como a la vista para hacer más livianas algunas tareas.

Controlador: Es el eje central de nuestra arquitectura, encargada de gestionar todas las peticiones, validar los inputs recibidos y dirigir cualquier petición de cualquier tipo. Solo se comunica con el modelo y responde a través de vistas.

Vista: Es la respuesta de cada controlador y lo que se le presenta al usuario final, se puede comunicar con el controlador, los “helpers” y el modelo (en algunas ocasiones).

1.6.5 My SQL

Gestor de Base de datos MySQL:

MySQL es uno de los sistemas gestores de bases de datos más populares. Es muy rápido, fiable y fácil de usar, surge para manipular bases de datos muy grandes. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas muy seguro que permite la autenticación básica para el acceso al servidor”. (DuBois 2012), (López 2005),(MySQL 2004).

Soporte grandes bases de datos. Ya es probable el uso de MySQL Server con bases de datos que contienen 50 millones de registros. También se conoce a

usuarios que usan MySQL server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.

Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, o 2003), los clientes pueden usar named pipes para la conexión. Basado en la experiencia personal del autor de este trabajo se decide emplear este SGBD para el desarrollo de la aplicación web.

La utilidad que tienen las aplicaciones **WAMP** se centra en:

-Facilidad de uso: La mayoría de soluciones vienen configuradas de tal manera que es muy sencillo arrancarlas y empezar a trabajar. Además, la mayoría de ellas se gestionan por medio de una interfaz mucho más intuitiva que un fichero editable.

-Configuración: El proceso de configurar, tanto cada una de las aplicaciones como el paquete WAMP es automático. De hecho, en la mayoría de los casos se configura automáticamente teniendo en cuenta la unidad donde se ha almacenado la carpeta que contiene el software. Y, como es evidente, también es posible editar dicha configuración de forma manual. Almacenado la carpeta que contiene el software. Y, como es evidente, también es posible editar dicha configuración de forma manual. (Bourdon 2012)

1.6.6 Framework Bootstrap

Framework de twitter para desarrollo de aplicaciones web. Sencillo y ligero. Puede bastar con un fichero CSS y uno JavaScript. Basado en los últimos estándares de desarrollo de Web HTML5, CSS3 y JavaScript/JQuery. Plugins de jQuery para validar entrada de datos, visualización tablas, grafos, etc. Curva de aprendizaje baja. Compatible con todos los navegadores habituales. (Carpenter and Bithell 2010),(Simmons-Duffin 2015)

1.6.7 Framework Symfony 2:

Ha sido ideado para exprimir al límite todas las nuevas características de PHP 5.3 y por eso es uno de los frameworks PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en el proyecto. También es el que más ideas incorpora del resto de frameworks, incluso de aquellos que no están programados con PHP como es Ruby On Rails, django o Spring.(Vilas Usó 2015), (Eguiluz 2007).

1.6.8 Lenguajes de Programación

PHP se trata de un lenguaje de programación interpretado en el servidor (server-side scripting). Fue desarrollado por el programador de origen danés Rasmus Lerdorf en 1994 con el propósito de facilitar el diseño de páginas web de carácter dinámico.(Heurtel 2011)

Se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en php son embebidos directamente en el código html y ejecutado por el servidor web a través de un intérprete antes de transferir al cliente que lo ha solicitado un resultado en forma de código html puro.(Holzner and Holzner 2009)

Para la realización de software será utilizado este lenguaje debido a que es:

Muy fácil de aprender.

Se caracteriza por ser un lenguaje muy rápido.

Soporta en cierta medida la orientación a objeto. Clases y herencia.

Es un lenguaje multiplataforma: Linux, Windows, entre otros.

Capacidad de conexión con la mayoría de los manejadores de base de datos:

MySQL, PostgreSQL, MS SQL Server, entre otras.

Posee documentación en su página oficial la cual incluye descripción y ejemplos

de cada una de sus funciones.

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Incluye gran cantidad de funciones.

HTML (Hypertext Markup Language) es un lenguaje sencillo con enlaces que conducen a otros documentos o fuentes de información relacionadas. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado. Es uno de los lenguajes más utilizados a la hora de desarrollo web ya que responde a las necesidades de manipulación y maquetación de los elementos de la aplicación. (Álvarez 2012)

CSS (Cascade Style Sheet) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos

con HTML y XHTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. (Pérez 2014)

Es un mecanismo simple para añadir estilos (ej., color, tipos de letras, formas) a documentos web. Esto trae como ventaja la accesibilidad del documento, mejora la claridad y reduce la complejidad de su mantenimiento. (Quijado, Quijado et al. 2012)

Java Script es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Java Script podemos crear efectos especiales en las páginas y definir interactividades con el usuario.

Java Script nos permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. Además, Java Script pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (McFarland 2011)

1.7 Conclusiones parciales del capítulo

Durante este capítulo se consolidó la base teórica con vistas al desarrollo del software mediante un estudio del estado de las principales herramientas y tecnologías que se proponen para el proyecto. Se profundiza en el análisis de las herramientas existentes hasta el momento, para concluir con los resultados que se esperan alcanzar al final del trabajo. Se define el modelo de desarrollo a seguir, el cual describe cada artefacto a obtener en cada etapa del desarrollo de esta aplicación y se sientan las bases que dan pie a su inicio. Queda plasmada de manera clara la necesidad de elaborar una herramienta informática para facilitar la gestión de los expedientes laborales en el departamento de recursos humanos de la universidad.

Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología de programación extrema XP. Se realiza un análisis detallado de las tecnologías a utilizar optando por utilizar el lenguaje de programación PHP con el framework Symfony y el gestor de bases de datos MySQL.

Capítulo II: Descripción de la solución propuesta

2.1 Introducción

En el presente capítulo se presenta un estudio detallado sobre la solución del proyecto relativo al proceso de gestión de los expedientes laborales en el Departamento de Recursos Humanos de la Universidad de Matanzas, apoyándose en la metodología XP, donde se va describiendo a través de las historias de usuarios y los actores que intervienen en el negocio; donde se tiene en cuenta para su correcto funcionamiento las tareas a desarrollar y la captura de las necesidades del cliente.

2.2 Descripción de la solución propuesta

Un requerimiento funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas. Como se define en la ingeniería de requerimientos, los requerimientos funcionales establecen los comportamientos del sistema.

La aplicación para la gestión de los expedientes laborales en el Departamento de Recursos Humanos de la Universidad de Matanzas pretende agilizar los procedimientos del trabajo, brindándoles a los técnicos del local la posibilidad de manipular la información a través de una aplicación simple y fácil de comprender. Para esto cada uno de ellos contará con un usuario que, una vez logueado en el sitio, les permita, de acuerdo a su rol, llevar a cabo las tareas pertinentes.

2.3 Requisitos Funcionales

Los requerimientos funcionales de un software son las respuestas del sistema a la acción de los usuarios. A continuación, se expondrán los requisitos funcionales referentes al proyecto:

1. Diseño de la interfaz de usuario
2. Diseño y Creación de la Base de Datos.
3. Autenticarse.
4. Gestionar Usuarios.
5. Gestionar Expedientes
6. Gestionar Escalas de Salario
7. Gestionar categorías docentes
8. Gestionar Bloques UMCC
9. Gestionar Grados científicos
10. Gestionar Consejo Popular
11. Gestionar Color de piel
12. Gestionar militancia
13. Gestionar Provincias

14. Gestionar Municipios
15. Gestionar Áreas
16. Gestionar Carreras
17. Gestionar idiomas
18. Gestionar País Graduado
19. Gestionar Organismos

2.4 Requisitos no Funcionales

Los requerimientos no funcionales son requisitos que especifican las propiedades del sistema, como restricciones del entorno o de implementación, rendimiento, dependencia de la plataforma, mantenibilidad, extensibilidad o fiabilidad. Son requisitos que especifican restricciones físicas sobre un requisito funcional. (Sampieri, Collado et al. 2009)

Apariencia o interfaz externa:

El diseño de la interfaz debe ser profesional, agradable, ágil, simple de usar. Se debe mostrar un contenido legible y confiable.

Usabilidad:

El sistema podrá ser usado por cualquier persona que posea los conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general, brindándole a cada momento al usuario diversas formas de acceder a las diferentes opciones que cuenta la web.

Rendimiento:

Dado a que el sistema creado es una aplicación Web, el tiempo de respuesta debe ser lo más cercano posible al tiempo real, se necesita un alto grado de eficiencia y un tiempo de respuesta muy breve, para lograr un incremento de la productividad.

Seguridad:

El sitio está creado para que solo el personal del Departamento de Recursos Humanos trabaje en él, en vista de la importancia de los datos que maneja. La información almacenada será revisada a través de validaciones que eliminen la entrada de datos irreales. Las contraseñas para los usuarios se deben encriptar, para almacenarlas en la base de datos, pues de esta manera se asegura que obtenerlas en texto claro, sea más difícil y quizás imposible.

Confiability:

El sistema debe garantizar el tratamiento adecuado de la información y la preservación de su integridad.

Portabilidad:

El sistema podrá implantarse sobre Windows, Linux o Unix de forma tal que no haya dificultad en cambiar, de una a otra plataforma, sin necesidad de efectuar cambios. Lo anterior se debe a que la aplicación está implementada sobre PHP, que es un lenguaje multiplataforma.

Requerimientos de software:

Para la implantación del sistema se requiere de:

Un servidor Unix, Linux o Windows, Apache HTTP Server como servidor Web.

Servidor de Base de Datos: MySQL 5.6.

PHP 5.5 o superior.

Del lado del cliente, Navegador de Internet pueden ser Microsoft Internet Explorer (a partir de la versión 3.0), Mozilla Firefox, Google Chrome entre otros.

Requerimientos de hardware:

Para el servidor:

Procesador Pentium 800MHz o superior, Pentium III, 128Mb de memoria RAM.

Las mencionadas condiciones están en dependencia de la cantidad de usuarios conectados, por lo que las características descritas son las mínimas.

Para el cliente:

Display con resolución 800 x 600.

Procesador Pentium II o superior con 64 MB de RAM como mínimo.

Microsoft Windows NT como sistema operativo. o superior

Las máquinas clientes deben tener acceso al servidor.

2.5 Integrantes del equipo

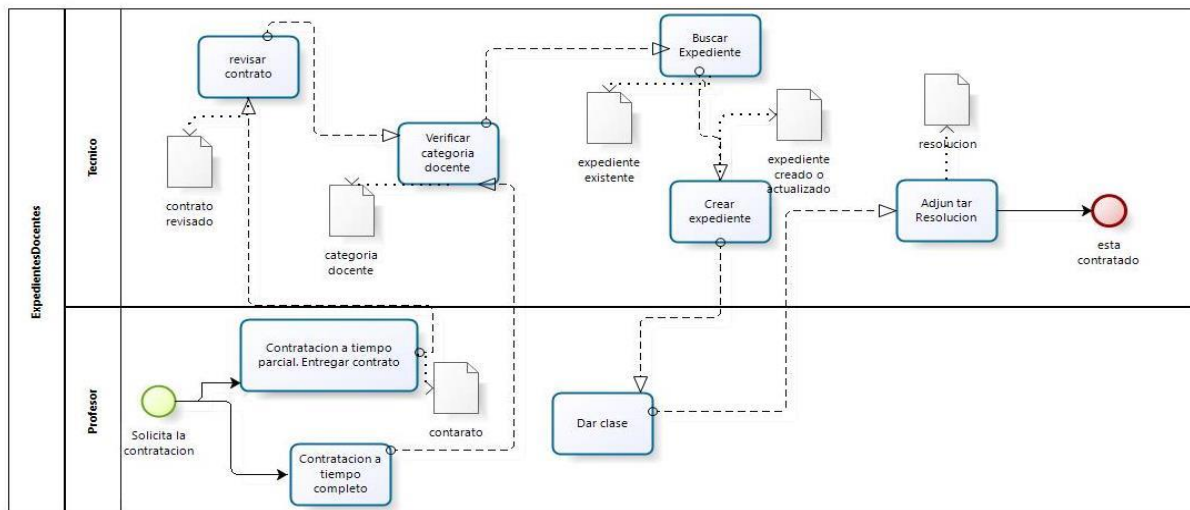
Tabla 1 Roles

Miembros	Roles	Metodología empleada
Departamento de Recursos Humanos de la Universidad de Matanzas	Cliente y consultor	XP
Noel Pérez Suria	Programador, encargado de pruebas	XP

Noel Pérez Suria	Gestor, Analista	XP
------------------	------------------	----

2.6 Diagrama de Procesos de Negocio

Permite el modelado de procesos de negocio, en un formato de flujo de trabajo.



Powered by
bizagi
Modeler

2.7 Historias de Usuarios

Las HU son la técnica utilizada en XP para especificar los requisitos del software. 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.

Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario. Estas ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. (Campos and Martínez 2015)

A continuación se muestran las escalas equivalentes a la prioridad en el negocio:

Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.

Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.

Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo. (Campos and Martínez 2015)

Escala Nominal de Riesgo en Desarrollo:

Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

2.7.1 Resumen de Historias de Usuario

Tabla 2 Resumen de Historias de Usuario

No	Nombre de Historias de usuario	Prioridad	Riesgo	Puntos Estimados	Iteración	Entrega
1	Diseño de la interfaz de usuario	Alta	Alto	2	1	
2	Diseño y creación de la base de datos	Alta	Alto	2	1	
3	Autenticarse	Alta	Alto	1	1	
4	Gestionar Usuario	Alta	Medio	1	1	
5	Gestionar Expediente	Alta	Medio	1	1	
6	Gestionar Escalas de Salario	Alta	Medio	1	1	
7	Gestionar Categoría Docente	Baja	Medio	1	2	
8	Gestionar Bloque UMCC	Baja	Medio	1	2	

9	Gestionar Grado Científico	Baja	Medio	1	2	
10	Gestionar Consejo Popular	Media	Medio	1	2	
11	Gestionar Color de Piel	Baja	Medio	1	2	
12	Gestionar Militancia	Baja	Medio	1	2	
13	Gestionar Provincias	Baja	Medio	0.5	3	
14	Gestionar Municipios	Baja	Medio	0.5	3	
15	Gestionar Área	Media	Medio	1	3	
16	Gestionar Carreras	Baja	Medio	1	3	
17	Gestionar Idiomas	Baja	Medio	1	3	
18	Gestionar País Graduado	Baja	Medio	1	3	
19	Gestionar Organismos	Baja	Medio	1	3	
20	Generar Reportes	Alta	Medio	3	3	
	Total			23		

Historias de usuario

Tabla 3 HU 1

Historia de usuario	
Número: 1	Usuario: Admin
Nombre de la Historia: Diseño de la interfaz de usuario	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 2	Iteraciones asignadas: 1
Programador Responsable: Noel Pérez Suria	
Descripción: Se muestra la interfaz de los diferentes usuarios.	
Observaciones: Confirmado por el cliente	

Tabla 4 HU 2

Historia de usuario	
Número: 2	Usuario: Admin
Nombre de la Historia: Diseño y creación de la base de datos	
Prioridad del Negocio: Media	Riesgo en Desarrollo: Medio
Puntos estimados: 2	Iteraciones asignadas: 1
Programador Responsable: Noel Pérez Suria	
Descripción: Se inicia cuando se realiza el diseño y creación de la base de datos.	
Observaciones: Confirmado por el cliente	

Tabla 5 HU 3

Historia de usuario	
Número: 3	Usuario: Todos
Nombre de la Historia: Autenticarse	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 1	Iteraciones asignadas: 1
Programador Responsable: Noel Pérez Suria	
Descripción: Se inicia cuando el administrador, el especialista y el trabajador intentan entrar al sistema para poder realizar las actividades asignadas de acuerdo con los permisos asignados.	
Observaciones: Confirmado por el cliente	

Tabla 6 HU 4

Historia de usuario	
Número: 4	Usuario: Admin
Nombre de la Historia: Gestionar usuarios	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Medio

Puntos estimados: 1	Iteraciones asignadas: 1
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el administrador decide insertar, modificar o eliminar a un usuario.	
Observaciones: Confirmado por el cliente	

Tabla 7 HU 5

Historia de usuario	
Número: 5	Usuario: Especialista
Nombre de la Historia: Gestionar expediente	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 1
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un expediente laboral.	
Observaciones: Confirmado por el cliente	

Tabla 8 HU 6

Historia de usuario	
Número: 6	Usuario: Especialista
Nombre de la Historia: Gestionar escalas de salario	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 1
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar una escala de salario.	
Observaciones: Confirmado por el cliente	

Tabla 9 HU 7

Historia de usuario	
Número: 7	Usuario: Especialista
Nombre de la Historia: Gestionar categoría docente	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar una categoría docente.	
Observaciones: Confirmado por el cliente	

Tabla 10 HU 8

Historia de usuario	
Número: 8	Usuario: Especialista
Nombre de la Historia: Gestionar Bloque UMCC	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un bloque.	
Observaciones: Confirmado por el cliente	

Tabla 11 HU 9

Historia de usuario	
Número: 9	Usuario: Especialista
Nombre de la Historia: Gestionar Grado científico	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2

Programador Responsable: Noel Pérez Suria
Descripción: Cuando el especialista decide insertar, modificar o eliminar un grado científico.
Observaciones: Confirmado por el cliente

Tabla 12 HU 10

Historia de usuario	
Número: 10	Usuario: Especialista
Nombre de la Historia: Gestionar Consejo Popular	
Prioridad del Negocio: Medio	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un consejo popular.	
Observaciones: Confirmado por el cliente	

Tabla 13 HU 11

Historia de usuario	
Número: 11	Usuario: Especialista
Nombre de la Historia: Gestionar Color de Piel	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un color de piel.	
Observaciones: Confirmado por el cliente	

Tabla 14 HU 12

Historia de usuario	
Número: 12	Usuario: Especialista
Nombre de la Historia: Gestionar Militancia	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un tipo de militancia.	
Observaciones: Confirmado por el cliente	

Tabla 15 HU 13

Historia de usuario	
Número: 13	Usuario: Especialista
Nombre de la Historia: Gestionar Provincias	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 0,5	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar una provincia.	
Observaciones: Confirmado por el cliente	

Tabla 16 HU 14

Historia de usuario	
Número: 14	Usuario: Especialista
Nombre de la Historia: Gestionar Municipios	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 0,5	Iteraciones asignadas: 3

Programador Responsable: Noel Pérez Suria
Descripción: Cuando el especialista decide insertar, modificar o eliminar un municipio.
Observaciones: Confirmado por el cliente

Tabla 17 HU 15

Historia de usuario	
Número: 15	Usuario: Especialista
Nombre de la Historia: Gestionar Área	
Prioridad del Negocio: Media	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un área.	
Observaciones: Confirmado por el cliente	

Tabla 18 HU 16

Historia de usuario	
Número: 16	Usuario: Especialista
Nombre de la Historia: Gestionar Carreras	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar una carrera.	
Observaciones: Confirmado por el cliente	

Tabla 19 HU 17

Historia de usuario

Número: 17	Usuario: Especialista
Nombre de la Historia: Gestionar Idiomas	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un idioma.	
Observaciones: Confirmado por el cliente	

Tabla 20 HU 18

Historia de usuario	
Número: 18	Usuario: Especialista
Nombre de la Historia: Gestionar País Graduado	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un país.	
Observaciones: Confirmado por el cliente	

Tabla 21 HU 19

Historia de usuario	
Número: 19	Usuario: Especialista
Nombre de la Historia: Gestionar Organismos	
Prioridad del Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el especialista decide insertar, modificar o eliminar un organismo.	

Observaciones: Confirmado por el cliente

Tabla 22 HU 20

Historia de usuario	
Número: 20	Usuario: Todos
Nombre de la Historia: Generar Reportes	
Prioridad del Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 3	Iteraciones asignadas: 3
Programador Responsable: Noel Pérez Suria	
Descripción: Cuando el administrador, especialista o trabajador decide generar un reporte para consultar información específica	
Observaciones: Confirmado por el cliente	

2.8 Estimación por puntos de función

Puntos de Función

Es una métrica que permite traducir en un **número** el tamaño de la funcionalidad que brinda un producto de software desde el punto de vista del usuario, a través de una suma ponderada de las características del producto.

Componentes:

EI: Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno.

EO: Procesos en los que se envía datos al exterior de la aplicación.

EQ: Procesos consistentes en la combinación de una entrada y una salida, en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

ILF: Grupos de datos relacionados entre sí internos al sistema.

EIF: Grupos de datos que se mantienen externamente.

Componentes	Cantidad	de	Total
	Componentes	por	su
	Peso		

EI (Entradas externas)	32* 3	96
EO (Salidas externas)	6 * 4	24
EQ (Consultas)	18* 3	54
ILF (Ficheros Lógicos Internos)	16 * 10	160
ILF (Ficheros Lógicos Externos)	0	0

Cálculo de los Puntos de Función sin Ajustar (PFSA)

Los PFSA se calculan como la suma de los productos de cada componente por su peso determinado en la tabla correspondiente.

$$PFSA = PFTe + PFTo + PFTq + PFtif + PFTef$$

$$PFSA = 96 + 24 + 54 + 160 + 0 = 334$$

Cálculo de los Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0.65 + (0.01 * ACT)]$$

Cálculo del Ajuste de Complejidad Técnica (ACT)

Para calcular el ACT se le va dando un valor entre 0 y 5 a cada Factor de Ajuste. Cuando cada Factor tenga un valor, se suman todos y así obtenemos el ACT.

No.de Factor	Factor de Ajuste	0<valor<5
1	Comunicación de Datos	2
2	Proceso Distribuido	4
3	Objetivos de Rendimiento	3
4	Configuración de Explotación Compartida	2
5	Tasa de transacciones	2
6	Entrada de Datos en Línea	2
7	Eficiencia con el Usuario Final	3
8	Actualizaciones en Línea	1

9	Lógica de Proceso Interno Compleja	3
10	Reusabilidad del Código	2
11	Conversión e Instalación contempladas	2
12	Facilidad de Operación	3
13	Instalaciones Múltiples	2
14	Facilidad de Cambios	2
Total	Ajuste de Complejidad Técnica(ACT)	33

$$PFA=PFSA*[0.65+(0.01*ACT)]$$

$$PFA=334*[0.65+(0.01*33)]$$

$$PFA=334*0.98$$

$$PFA=327.32$$

Cálculo del Esfuerzo

Líneas de Código (LC)

$$LC = PFA * (\text{Líneas por PF})$$

$$LC = 327.32 * 20$$

$$LC = 6546.4$$

Esfuerzo en Horas/Personas (E)

$$E=PFA/ (1/8 \text{ persona/hora})$$

$$E=327.32/0.125$$

$$E=2618.56 \text{ horas/personas}$$

Tomando 24 días laborales en el mes y 8 horas productivas al día se obtiene

192 horas laborables al mes

Duración del proyecto en meses

$$2618.56 \text{ horas/personas}/1 \text{ persona}=2618.56 \text{ horas}$$

DM=2618.56 horas/192 horas=13.64 meses aproximadamente 14 meses por el margen de error de la estimación por puntos de función.

Costo total del proyecto

$$CT=\text{sueldo de 1 persona} * \text{cantidad de personas} * DM$$

$$CT=250*1*14$$

CT=3500

Este cálculo permite al autor definir el plan de entregas y de iteraciones como se mostrará en el epígrafe que sigue.

2.9 Planificación de Iteraciones

Las iteraciones fueron planificadas en entrevistas con el cliente, cada una incluye la realización de una o más historias de usuarios y pueden realizarse tantas como sean necesarias. Al final de cada iteración no es necesario realizar una entrega aunque se recomienda que las entregas se realicen una vez concluidas las iteraciones a realizar. Para realizar la planificación se tuvo en cuenta que las iteraciones no fuesen muy largas y no excedieran las tres semanas.

2.10 Plan de entrega

Las entregas del sistema están compuestas por una iteración, aunque pueden estar compuestas por una o más. Realizar una planificación para las entregas hace que se pueda realizar un control por parte del plan acordado con el cliente.

2.11 Tareas de Iteración

Tabla 24 Resumen de Tareas de Iteración

No	Nombre de la Historia de usuario	No de Iteración	Tareas Asignadas
1	Diseño de la interfaz de usuario	1	1-Interfaz Principal
			2-Otras Interfaces
2	Diseño y creación de la base de datos	1	3-Diseño de la base de datos
			4-Creación de la base de datos
3	Autenticarse	1	5-Autenticarse
			6-Validar usuario
4	Gestionar usuarios	1	7-Insertar usuario
			8-Editar usuario

			9-Eliminar usuario
5	Gestionar Expediente	1	10-Ver expedientes
			11-Insertar expediente
			12-Editar expediente
			13-Eliminar expediente
6	Gestionar escalas de salario	1	14-Ver escalas de salario
			15-Insertar escala de salario
			16-Editar escala de salario
			17-Eliminar escala de salario
7	Gestionar categorías docentes	2	18-Insertar categoría docente
			19-Editar categoría docente
			20-Eliminar categoría docente
8	Gestionar Bloque UMCC	2	21-Insertar Bloque UMCC
			22-Editar Bloque UMCC
			23-Eliminar Bloque UMCC
9	Gestionar Grado científico	2	24-Insertar Grado científico
			25-Editar Grado científico
			26-Eliminar Grado científico
10	Gestionar Consejo Popular	2	27-Ver Consejos Populares
			28-Insertar Consejo Popular
			29-Editar Consejo Popular
			30-Eliminar Consejo Popular
11	Gestionar Color de piel	2	31-Insertar Color de piel
			32-Editar Color de piel
			33-Eliminar Color de piel

12	Gestionar Militancia	2	34-Insertar militancia
			35-Editar militancia
			36-Eliminar militancia
13	Gestionar Provincias	3	37-Insertar provincia
			38-Editar provincia
			39-Eliminar provincia
14	Gestionar Municipios	3	40-Insertar municipio
			41-Editar municipio
			42-Eliminar municipio
15	Gestionar Área	3	43-Insertar área
			44-Editar área
			45-Eliminar área
16	Gestionar Carreras	3	46-Insertar carrera
			47-Editar carrera
			48-Eliminar carrera
17	Gestionar Idiomas	3	49-Insertar idioma
			50-Editar idioma
			51-Eliminar idioma
18	Gestionar País Graduado	3	52-Insertar país
			52-Editar país
			53-Eliminar país
19	Gestionar Organismos	3	54-Insertar organismo
			55-Editar organismo
			56-Eliminar organismo
20	Generar Reportes	3	57-Generar reportes

A continuación se relacionan algunas tareas de iteración que tenían mayor peso en el desarrollo de este proyecto. El resto podrán encontrarse en la documentación del sistema:

Tabla 25 Tarea 1 HU 1

Tarea	
Número de tarea: 1	Número de historia: 1
Nombre de Tarea: Interfaz Principal	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Crear la interfaz visual principal de la aplicación.	

Tabla 26 Tarea 4 HU 2

Tarea	
Número de tarea: 4	Número de historia: 2
Nombre de Tarea: Creación de la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Se crea la base de datos empleando my SQL	

Tabla 27 Tarea 5 HU 3

Tarea	
Número de tarea: 5	Número de historia: 3
Nombre de Tarea: Autenticarse	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Dado un usuario y su contraseña se autentifica en el sistema y se	

le otorgan los permisos respectivos a su rol.

Tabla 28 Tarea 7 HU 4

Tarea	
Número de tarea: 7	Número de historia: 4
Nombre de Tarea: Insertar usuario	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Permite insertar un usuario con sus permisos correspondientes.	

Tabla 29 Tarea 10 HU 5

Tarea	
Número de tarea: 10	Número de historia: 5
Nombre de Tarea: Ver expediente	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Cuando un especialista decide ver un expediente existente.	

Tabla 30 Tarea 11 HU 5

Tarea	
Número de tarea: 11	Número de historia: 5
Nombre de Tarea: Insertar expediente	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Cuando un especialista decide insertar un nuevo expediente.	

Tabla 31 Tarea 12 HU 5

Tarea	
Número de tarea: 12	Número de historia: 5
Nombre de Tarea: Editar expediente	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Cuando un especialista decide modificar un expediente existente.	

Tabla 32 Tarea 13 HU 5

Tarea	
Número de tarea: 13	Número de historia: 5
Nombre de Tarea: Eliminar expediente	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Cuando un especialista decide eliminar un expediente existente.	

Tabla 33 Tarea 14 HU 6

Tarea	
Número de tarea: 14	Número de historia: 6
Nombre de Tarea: Ver Escala de salario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando un especialista decide ver una escala de salario existente.	

Tabla 34 Tarea 15 HU 6

Tarea	
Número de tarea: 15	Número de historia: 6
Nombre de Tarea: Insertar Escala de salario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando un especialista decide insertar una escala de salario.	

Tabla 35 Tarea 16 HU 6

Tarea	
Número de tarea: 16	Número de historia: 6
Nombre de Tarea: Editar Escala de salario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando un especialista decide modificar una escala de salario existente.	

Tabla 36 Tarea 17 HU 6

Tarea	
Número de tarea: 17	Número de historia: 6
Nombre de Tarea: Eliminar Escala de salario	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Noel	
Descripción: Cuando un especialista decide eliminar una escala de salario.	

Tabla 31 Tarea 27 HU 10

Tarea	
Número de tarea: 27	Número de historia: 10
Nombre de Tarea: Ver Consejo Popular	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando un especialista decide ver un consejo popular existente.	

Tabla 32 Tarea 28 HU 10

Tarea	
Número de tarea: 28	Número de historia: 10
Nombre de Tarea: Insertar Consejo Popular	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando el especialista decide añadir un consejo popular.	

Tabla 33 Tarea 29 HU 10

Tarea	
Número de tarea: 29	Número de historia: 10
Nombre de Tarea: Editar Consejo Popular	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando el especialista decide modificar un consejo popular existente.	

Tabla 34 Tarea 30 HU 10

Tarea	
Número de tarea: 30	Número de historia: 10
Nombre de Tarea: Eliminar Consejo Popular	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando el especialista decide eliminar un consejo popular existente.	

Tabla 35 Tarea 57 HU 20

Tarea	
Número de tarea: 57	Número de historia: 20
Nombre de Tarea: Generar reporte	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Noel	
Descripción: Cuando se decide generar algún reporte de la información existente.	

2.12 Diseño de Tarjetas de Clase, Responsabilidad y Colaboración

Una de las principales funcionalidades que tiene las tarjetas CRC es mostrar las colaboraciones que se establecen entre las clases que componen el sistema, donde cada tarjeta representa una clase con su nombre en la parte superior. A continuación, se muestran algunas de las tarjetas que se crearon las otras se encuentran en el la Documentación del Sistema.

Tarjetas CRC
Clase: usuario
Superclase:

Subclase:	
Descripción: En esta clase se guarda todo lo relacionado con los datos de los usuarios del sistema.	
Atributos	
Id	Int(11)
Usuario	Varchar(255)
Password	Varchar(255)
Rol	Int(11)

Tarjetas CRC	
Clase: categoría_docente	
Superclase:	
Subclase:	
Descripción: En esta clase se guarda todo lo relacionado con los datos de las categorías docentes de los profesores.	
Atributos	
Id_categoría	Int(11)
categoría	Varchar(255)

2.13 Análisis de los Costos(Modelo matemático COCOMO II)

El modelo de estimación de software COCOMO II fue propuesto y elaborado por Barry Boehm, este modelo es uno de los modelos de estimación más utilizados y con una gran cantidad de documentación. El modelo permite obtener el esfuerzo y el tiempo que se requiere en un proyecto de software a partir de una medida del tamaño del mismo expresada en el número de líneas de código que se estimen generar para la creación del producto.

Este modelo consta de tres etapas que se presentan a continuación:
(Pressman, 2010)

- **Modelo de composición de aplicación:** se usa en las primeras etapas de proyecto, cuando son primordiales la elaboración de prototipos de las interfaces de usuario, la consideración de la interacción del software y el sistema, la valoración del rendimiento y la evaluación de la madurez de la tecnología.
- **Modelo de etapa temprana de diseño:** Se usa una vez estabilizados los requisitos y establecida la arquitectura básica del software.
- **Modelo de etapa postarquitectónica:** Se usa durante la construcción del software.

En este proyecto se realizó mediante el **modelo de etapa temprana de diseño**. En esta etapa se tiene poca información, se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o aspectos específicos del proceso a utilizar. Ese nivel de detalle en este modelo es consistente con el nivel general de información disponible y con nivel general de estimación detallada que es necesaria en estas etapas, lo que concuerda con el uso de Puntos de Función. Para estimar tamaño usa Puntos de Función No Ajustados como métrica de medida y un número reducido de factores de costo.

Conclusiones

Se realiza una descripción de las HU que permitió guiar al equipo durante el proceso de desarrollo de la solución propuesta. Se define el plan principal de iteraciones y el plan de entregas a seguir, siempre teniendo en cuenta que las mismas pueden sufrir modificaciones durante el proceso de implementación.

Capítulo III: Resultados del Trabajo Desarrollado. Elementos de la Validación Práctica de la Propuesta de Solución del Problema Científico.

3.1 Introducción

En el presente capítulo se explicarán los casos de prueba, las estrategias de prueba, las pruebas de aceptación y el análisis de los resultados obtenidos. Se mostrará también el análisis de factibilidad y costo. El desarrollo de un software conlleva a un conjunto de actividades de producción en las que las posibilidades del error humano son muy posibles, éstos pueden ser por un mal enfoque de los objetivos, o por un diseño y desarrollo erróneo. Como resultado del capítulo II se presenta un software que responde a los objetivos planteados en la introducción de este trabajo, y este capítulo estará dedicado esencialmente a la validación dicho software.

3.2 Pruebas

3.2.1 Plan de Pruebas

No	Historias de usuarios	Pruebas a realizar
1	Diseño de la interfaz de usuario	Test de conexión
2	Diseño y creación de la base de datos	Test de Base de datos
3	Autenticarse	Test de autenticarse
4	Gestionar usuarios	Test de listar usuarios. Test de insertar usuario. Test de editar usuario. Test de eliminar usuario
5	Gestionar expedientes	Test de listar expedientes Test de insertar expediente Test de editar expediente Test de eliminar expediente
6	Gestionar escalas de salario	Test de listar escala de salario Test de insertar escala de salario

		<p>Test de editar escala de salario</p> <p>Test de eliminar escala de salario</p>
7	Gestionar categorías docentes	<p>Test de listar categoría docente</p> <p>Test de insertar categoría docente</p> <p>Test de editar categoría docente</p> <p>Test de eliminar categoría docente</p>
8	Gestionar Bloques UMCC	<p>Test de insertar bloque UMCC</p> <p>Test de editar bloque UMCC</p> <p>Test de eliminar bloque UMCC</p>
9	Gestionar Grado científico	<p>Test de insertar grado científico</p> <p>Test de editar grado científico</p> <p>Test de eliminar grado científico</p>
10	Gestionar Consejo Popular	<p>Test de listar consejo popular</p> <p>Test de insertar consejo popular</p> <p>Test de editar consejo popular</p> <p>Test de eliminar consejo popular</p>
11	Gestionar Color de Piel	<p>Test de insertar color de piel</p> <p>Test de editar color de piel</p> <p>Test de eliminar color de piel</p>
12	Gestionar Militancia	<p>Test de insertar militancia</p> <p>Test de editar militancia</p> <p>Test de eliminar militancia</p>
13	Gestionar Provincias	<p>Test de insertar provincia</p> <p>Test de editar provincia</p> <p>Test de eliminar provincia</p>
14	Gestionar Municipios	<p>Test de insertar municipio</p>

		Test de editar municipio Test de eliminar municipio
15	Gestionar Áreas	Test de insertar área Test de editar área Test de eliminar área
16	Gestionar Carreras	Test de insertar carrera Test de editar carrera Test de eliminar carrera
17	Gestionar Idiomas	Test de insertar idioma Test de editar idioma Test de eliminar idioma
18	Gestionar País Graduado	Test de insertar país Test de editar país Test de eliminar país
19	Gestionar Organismos	Test de insertar organismo Test de editar organismo Test de eliminar organismo
20	Generar Reportes	Test de generar reportes

3.2.2 Pruebas de aceptación

Las Pruebas de Aceptación (PA) son las realizadas por el cliente y usuarios finales de la aplicación. En estas serán probadas las funcionalidades definidas por el cliente y descritas en las historias de usuario, además de los aspectos de seguridad requeridos. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso.

Tabla 36 HU 1

Caso de Prueba	
Número de caso de prueba:1	Número de historia de usuario:1
Nombre de Caso de Prueba: Test Diseño de la interfaz de usuario	
Descripción: Se muestra la interfaz correspondiente a la vista solicitada por el usuario.	
Condiciones de ejecución: Que la página esté conectada a la base de datos.	
Entradas: Vista del sistema.	
Resultado esperado: se muestra la interfaz correspondiente a la petición del usuario.	
Evaluación: Satisfactoria.	
Responsable: Noel Pérez Suria	

Ilustración 1 Prueba 1

Gestion de los Recursos H x

localhost/Recursos%20Humanos/web/app.php/index

Recursos Humanos Salir Hola! mary

Gestionar Expediente

Gestionar Escalas de Salarios

Gestionar Categoría Docente

Gestionar Bloque UMCC

Gestionar Grado científico

Gestionar Consejo Popular

Gestionar Color de Piel

Gestionar Militancia

Gestionar Provincias

Gestionar Municipios

Gestionar Área

Gestionar Carreras

Gestionar Idiomas

Gestionar País Graduado

Gestionar Organismos

Bienvenido Técnico : mary




Tabla 37 HU 2

Caso de Prueba	
Número de caso de prueba:2	Número de historia de usuario:2
Nombre de Caso de Prueba: Test Diseño y creación de la base de datos	
Descripción: Verifica el funcionamiento de la base de datos	
Condiciones de ejecución: Estar conectado a la base de datos.	
Entradas: Valores para leer o escribir en la base de datos	
Resultado esperado: se muestran o guardan los datos correctamente.	
Evaluación: Satisfactoria.	
Responsable: Noel Pérez Suria	

Ilustración 2 Prueba 2

Gestion de los Recursos H x

localhost/Recursos%20Humanos/web/app.php/consejo_popular/

Recursos Humanos Reporte Salir Hola! mary

- Gestionar Expediente
- Gestionar Escalas de Salarios
- Gestionar Categoría Docente
- Gestionar Bloque UMCC
- Gestionar Grado científico
- Gestionar Consejo Popular
- Gestionar Color de Piel
- Gestionar Militancia
- Gestionar Provincias
- Gestionar Municipios
- Gestionar Área
- Gestionar Carreras
- Gestionar Idiomas
- Gestionar País Graduado
- Gestionar Organismos

Lista de Consejos Populares

Código del Consejo	Consejo Popular	Provincia	Municipio	Acciones
1	Pueblo Nuevo	Matanzas	Matanzas	Mostrar Editar
2	La Marina	Matanzas	Cárdenas	Mostrar Editar
3	Diez De Octubre	Villa Clara	Remedio	Mostrar Editar

Nuevo

Tabla 38 HU 6

Caso de Prueba	
Número de caso de prueba:3	Número de historia de usuario:6
Nombre de Caso de Prueba: Test de Listar escalas de salario	
Descripción: Se permiten listar todas las escalas de salario existentes	
Condiciones de ejecución: Estar conectado a la base de datos.	
Entradas: Acceder a la vista donde se muestra la lista de escalas de salario existentes.	
Resultado esperado: Se deben mostrar en una tabla todas las escalas de salario almacenadas en la base de datos.	
Evaluación: Satisfactoria.	
Responsable: Noel Pérez Suria	

Ilustración 3 Prueba 3

Gestion de los Recursos H x

localhost/Recursos%20Humanos/web/app.php/escala_salario/

Recursos Humanos Reporte Salir Hola! mary

- Gestionar Expediente
- Gestionar Escalas de Salarios
- Gestionar Categoría Docente
- Gestionar Bloque UMCC
- Gestionar Grado científico
- Gestionar Consejo Popular
- Gestionar Color de Piel
- Gestionar Militancia
- Gestionar Provincias
- Gestionar Municipios
- Gestionar Área
- Gestionar Carreras
- Gestionar Idiomas
- Gestionar País Graduado
- Gestionar Organismos

Lista de Escalas

Id de Escala	Escala de Salario	Acciones
1	250 - 400	Mostrar Editar
2	500 - 750	Mostrar Editar
3	200 - 300	Mostrar Editar
8	100 - 200	Mostrar Editar

Nuevo

Tabla 39 HU 6

Caso de Prueba	
Número de caso de prueba:4	Número de historia de usuario:6
Nombre de Caso de Prueba: Test de Insertar escalas de salario	
Descripción: Verificar que todos los datos de las escalas pueden ser insertados y almacenados en la base de datos correctamente. También se insertarán datos incorrectos y se dejarán espacios en blanco para comprobar que estos no son aceptados.	
Condiciones de ejecución: Estar conectado a la base de datos.	
Entradas: <ul style="list-style-type: none"> -Insertar los datos correctamente. -Insertar los datos incorrectamente. -Dejar campos en blanco. 	

Resultado esperado: El sistema cuando se inserten los datos correctamente debe almacenarlos en la base de datos y mostrarlos. Cuando se inserten datos incorrectos o campos en blanco mostrar una alerta.
Evaluación: Satisfactoria.
Responsable: Noel Pérez Suria

Tabla 40 HU 6

Caso de Prueba	
Número de caso de prueba:5	Número de historia de usuario:6
Nombre de Caso de Prueba: Test de Editar escalas de salario	
Descripción: Verificar que todos los datos de las escalas pueden ser modificados y almacenados en la base de datos correctamente. También se modificarán datos de manera incorrecta y se dejarán espacios en blanco para comprobar que estos no son aceptados.	
Condiciones de ejecución: Estar conectado a la base de datos.	
Entradas: <ul style="list-style-type: none"> -Editar los datos correctamente. -Editar los datos incorrectamente. -Dejar campos en blanco. 	
Resultado esperado: El sistema cuando se editen los datos correctamente debe almacenarlos en la base de datos y mostrarlos. Cuando se editen datos incorrectos o campos en blanco mostrar una alerta.	
Evaluación: Satisfactoria.	
Responsable: Noel Pérez Suria	

Tabla 41 HU 6

Caso de Prueba	
Número de caso de prueba:6	Número de historia de usuario:6
Nombre de Caso de Prueba: Test de Eliminar escalas de salario	
Descripción: Verificar que el registro deseado puede ser eliminado de la tabla.	

Condiciones de ejecución: Estar conectado a la base de datos.
Entradas: Se eliminará la escala de salario deseada por el usuario.
Resultado esperado: Se elimina la escala de salario de la base de datos.
Evaluación: Satisfactoria.
Responsable: Noel Pérez Suria

3.2.3 Pruebas Funcionales

- Insertar categoría docente

1-Atributos:

Id de la categoría

Nombre de la categoría

2- Clases de equivalencia por cada atributo

Tabla 42

Atributo	Válida	Representante	Inválida	Representante
Id de la categoría	1-Un valor numérico positivo.	10	2-Que esté vacío.	NULL
			3-Introducir letras o caracteres extraños.	qwegr
			4-Que el número sea negativo.	-10
Nombre	5- Combinación de letras que comience con	Profesor Auxiliar	6- Que esté vacío	NULL
			7- Que no comience con mayúscula	profesor auxiliar

	mayúscula.		8- Que contenga números o caracteres extraños	N34%
--	------------	--	---	------

3- Definir casos de prueba

Tabla 43

No	Clase de equivalencia	Id de categoría	Nombre	Resultado esperado
1	1,5	10	Profesor Auxiliar	Resultado esperado
2	1,6	10	NULL	El nombre no puede estar vacío.
3	1,7	10	Profesor Auxiliar	El nombre no puede comenzar con minúscula.
4	1,8	10	N34%	El nombre no puede contener caracteres extraños ni numéricos.
5	2,5	NULL	Profesor Auxiliar	El id no puede estar vacío.
6	3,5	qweqr	Profesor Auxiliar	El id no puede contener letras o caracteres extraños.
7	4,5	-10	Profesor Auxiliar	El id no puede ser negativo.

4- Tabla por casos de prueba

Tabla 44

No.	1	
Requerimiento	Insertar categoría docente	
Objetivo	Probar la acción de insertar categoría docente.(clases equivalentes 1,5)	
Tipo de Prueba	Funcional	
Hardware	Un procesador Core i5 , disco duro de 750GB, memoria RAM de 6GB	
Software	Sistema Operativo Windows 7 o superior, Base de Datos Xampp, Navegador Mozilla Firefox 3.5 o superior	
Personal	Ingeniero de Pruebas	
Casos de Prueba		
Datos de Entrada	Id de categoría: 10 Nombre: Profesor Auxiliar	
Resultados Esperados	La categoría se ingresa correctamente.	
Resultados Obtenidos	Si(X) No()	
Casos de Excepción:	Comentarios:	
Aprobado por: Noel Pérez Suria	Cargo: Líder de pruebas	Fecha:

Tabla 45

No.	2	
Requerimiento	Insertar categoría docente	
Objetivo	Probar la acción de insertar categoría docente.(clases equivalentes 1,6)	
Tipo de Prueba	Funcional	

Hardware	Un procesador Core i5 , disco duro de 750GB, memoria RAM de 6GB	
Software	Sistema Operativo Windows 7 o superior, Base de Datos Xampp, Navegador Mozilla Firefox 3.5 o superior	
Personal	Ingeniero de Pruebas	
Casos de Prueba		
Datos de Entrada	Id de categoría: 10 Nombre: NULL	
Resultados Esperados	La categoría no puede estar vacía	
Resultados Obtenidos	Si() No()	
Casos de Excepción:	Comentarios:	
Aprobado por: Noel Pérez Suria	Cargo: Líder de pruebas	Fecha:

-Editar categoría docente

1-Atributos:

Id de la categoría

Nombre de la categoría

2- Clases de equivalencia por cada atributo

Tabla 46

Atributo	Válida	Representante	Inválida	Representante
Id de la categoría	1-Que se encuentre en la base de datos	10	2-Que no esté en la base de datos.	NULL

Nombre	3- Que se encuentre en la base de datos	Profesor Auxiliar	4- Que no esté en la base de datos.	NULL
--------	---	-------------------	-------------------------------------	------

3- Definir casos de prueba

Tabla 47

No	Clase de equivalencia	Id de categoría	Nombre	Resultado esperado
1	1,3	10	Profesor Auxiliar	Resultado esperado
2	1,2	10	NULL	El nombre no se encuentra en la base de datos.
3	2,3	NULL	Profesor Auxiliar	El id no se encuentra en la base de datos.
4	2,4	NULL	NULL	El id y el nombre no se encuentran en la base de datos.

4- Tabla por casos de prueba

Tabla 48

No.	1
Requerimiento	Editar categoría docente
Objetivo	Probar la acción de editar categoría docente.(clases equivalentes 1,3)
Tipo de Prueba	Funcional
Hardware	Un procesador Core i5 , disco duro de 750GB, memoria RAM de 6GB
Software	Sistema Operativo Windows 7 o

	superior, Base de Datos Xampp, Navegador Mozilla Firefox 3.5 o superior
Personal	Ingeniero de Pruebas
Casos de Prueba	
Datos de Entrada	Id de categoría: 10 Nombre: Profesor Auxiliar
Resultados Esperados	La categoría se editó correctamente
Resultados Obtenidos	Si(X) No()
Casos de Excepción:	Comentarios:
Aprobado por: Noel Pérez Suria	Cargo: Líder de pruebas Fecha:

3.3 Análisis de los resultados obtenidos

La aplicación desarrollada presenta beneficios muy favorables ya que este software le ofrece una herramienta que facilita su trabajo y da solución a la problemática planteada. La aplicación les permite contar con un sistema de fácil acceso y el aumento de la capacidad de almacenamiento de la información permitiendo de esta forma aumenta la calidad de trabajo y por ende disminuye el esfuerzo del usuario para obtener la información necesaria. El usuario tendrá la posibilidad de trabajar con a información de una manera mucho más ágil, satisfaciéndose así las necesidades requeridas por el cliente.

Conclusiones parciales

En este capítulo se mostraron los elementos de prueba y los casos de prueba que se le aplicaron, además se explicaron las estrategias de prueba. Las pruebas se convierten en una herramienta de desarrollo, son un paso imprescindible mediante la verificación del buen funcionamiento de un

software. La planificación inicial se cumplió por lo que se pudo lograr una aplicación web con mucha manejabilidad y con buena apariencia. Se utilizaron varias de las herramientas más actuales para su desarrollo y el plan de entrega fue cumplido con éxito.

Conclusiones

Siendo consecuentes con los objetivos que se plantearon al inicio de esta investigación, se hizo un análisis para determinar los trabajos más importantes relacionados con la gestión de los recursos humanos en una entidad determinada.

Siguiendo esta idea se encontró la existencia del software ASSETS y su poderosa influencia, pero que aún así no daba solución al problema planteado en el departamento de la universidad. En mira de esto y para satisfacer el objetivo general se decidió desarrollar una aplicación web para la Gestión de los Expedientes Laborales en el Departamento de Recursos Humanos de la Universidad de Matanzas.

Se validó la propuesta de software, aplicando una serie de pruebas para determinar sus fallas y el nivel de calidad y usabilidad que presentaba, dándole

así solución al problema científico y la hipótesis que se habían planteado con el desarrollo de esta aplicación.

Recomendaciones

Desde el punto de vista del alcance del trabajo proponemos las siguientes recomendaciones:

- 1- Aprovechar las posibilidades que brinda el tema e implementar nuevas funcionalidades según se crea necesario.
- 2- Aumentar el número de reportes que brinde la aplicación para la satisfacción de los usuarios que la utilizan.

Bibliografías

Alvarez, R. (2012). "Introducción al HTML." Recuperado el 1.

Balaguera, Y. D. A. (2015). "Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual." Revista de Tecnología 12(2).

Bascón, E. (2010). "El patrón de diseño modelo-vista-controlador (MVC) y su implementación en Java-swing." Acta Nova 2(4): 493-507.

Bello, P. N. M. (2013). "Aplicación web para el modelado de expresiones faciales en un avatar."

Campos, S. G. and L. F. F. Martínez (2015). "Programación Extrema: Prácticas, Aceptación y Controversia." CULCyT(15).

- Celma Giménez, M., J. C. Casamayor Ródenas and L. Mota Herranz (2003). Bases de datos relacionales, Madrid [etc.]: Prentice-Hall.
- Eguiluz, J. (2007). Desarrollo web ágil con Symfony
- Espinosa, A. T., J. G. C. Sagredo, M. M. Reyes and M. d. L. L. García (2012).
- "Automatización de la codificación del patrón modelo vista controlador (MVC) en proyectos orientados a la Web." CIENCIA ergo-sum 19(3): 239-250.
- Gómez, R. and H. Cristina (2015). "Control territorial y resistencias: Una lectura desde la seguridad humana."
- González, Y. D. and Y. F. Romero (2012). "Patrón Modelo-Vista-Controlador." Revista Telem@tica 11(1): 47-57.
- Haddad, P. R. (2014). "Planificación participativa: conceptos y métodos operacionales."
- Revista EURE-Revista de Estudios Urbano Regionales 10(29).
- Heurtel, O. (2011). PHP 5.3: Desarrollar un sitio Web dinámico e interactivo, Ediciones ENI.
- Holzner, S. and S. Holzner (2009). PHP: manual de referencia.
- Letelier, P. (2010). "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)."
- McFarland, D. S. (2011). JavaScript & jQuery: the missing manual, " O'Reilly Media, Inc."
- Navarro, R. F. and C. A. R. Romero (2012). "Una interpretación del concepto de gestión del conocimiento de Nonaka y Takeuchi usando la ficción literaria/An interpretation of the concept of knowledge management proposed by Nonaka & Takeuchi using literary fiction*." Apuntes del CENES 31(54): 227.
- Pérez, J. E. (2014). "Introducción a CSS."
- Quijado, L., J. L. Quijado, J. Aracil, F. Aracil, F. Gordillo, B. B. B. B. Welch, A. Usatechi, J.
- M. M. Bueno, A. M. A. Usategui and A. del Moral Bueno (2012). Domine HTML 5 y CSS 2, Microsoft Corporation.
- Sánchez, T., Letelier and L.E.A (2003). Metodologías Ágiles en el desarrollo de Software.

Tracz, W. (2002). "Extreme Programming (XP)." Encyclopedia of Software Engineering.

Vilas Usó, P. (2015). "Desarrollo de una web corporativa basada en Symfony."

Zafra, A., E. Gibaja, M. Luque and S. Ventura (2014). "Diseño de aplicaciones cliente/servidor para el aprendizaje de las tecnologías de comunicación." Iniciación a la Investigación.