

**UNIVERSIDAD DE MATANZAS
SEDE: "CAMILO CIENFUEGOS"**



**Facultad de Ciencias Económicas e Informática
Departamento de Informática
Curso 2017-2018**

**Trabajo de Diploma en opción al título de
Ingeniero Informático**



**Tema de Investigación: Gestión del control
a las actividades docentes**

**Autor: Wuilliam Hernández Rodríguez
Tutor: Dr.C. Walfredo González Hernández**

**Matanzas, Cuba
Junio, 2018**

DEDICATORIA

Dedicatoria

Dedico esta investigación a todos aquellos que me han apoyado de manera incondicional a la realización de este trabajo y el haber depositado su confianza en mí.

A mi madre, hermano y novia.

AGRADECIMIENTOS

Agradecimientos

A mi madre y hermano a los cuales les debo todo lo que soy, por su apoyo y confianza, siempre les estaré eternamente agradecida.

A mi novia por apoyarme y reprenderme en los momentos que más lo necesitaba.

A mi familia que estuvieron siempre preocupados por mi desempeño en la tesis y que tuvieron que escucharme hablar durante meses de la misma.

A mi tutor Walfredo por su paciencia, ayuda y dedicación en todo momento.

A todos los profesores que han contribuido a mi formación en el transcurso de mi carrera.

A las amistades que me ayudaron en todo momento Ediel, Yonelkis, Isselis y los demás compañeros de aula.

A todo aquel que hizo posible que pudiera graduarme.

A todos Gracias

DECLARACIÓN DE AUDITORÍA

Declaración de autoría

Yo, Wuilliam Hernández Rodríguez, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, y en especial, a la Facultad de Ciencias Técnicas, al Departamento de Informática, a que hagan el uso que estimen pertinente de él.

Para que así conste, firmo la presente a los 27 días del mes de junio del 2018.

Firma del autor
Wuilliam Hernández Rodríguez

Firma del tutor
Ing. Walfredo

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Opinión del Tutor del Trabajo de Diploma

RESUMEN

Resumen

Este Trabajo presenta una Aplicación web para ayudar a la gestión de la información que se maneja en el proceso de control a las actividades docentes, la cual constituye una atractiva propuesta que contribuye y facilita este proceso tanto en nuestro departamento como en la Universidad en general. Para ello se empleó la metodología de desarrollo de software Programación Extrema, se utilizó el lenguaje de programación de ambiente web PHP, el *framework* Symfony, como gestor de base de datos MySQL y como servidor web se empleó Apache. Se presentan la metodología de la investigación empleada, las definiciones asociadas al objeto de estudio y al campo de acción.

SUMMARY

Summary

ÍNDICE

Índice

Capítulo 1: Marco teórico referencial	1
1.1 Introducción	1
1.2 Objeto de Estudio.....	1
1.2.1 Caracterización del proceso de Control a las Actividades Docentes. .	1
.....	2
1.3 Métodos de la investigación	3
1.3.1 Métodos teóricos empleados	3
1.3.2 Métodos empíricos empleados	4
1.4 Herramientas, tecnologías y metodologías de desarrollo	4
1.4.1 Tendencias tecnológicas a considerar	4
1.4.2 Metodologías de desarrollo.....	7
1.4.3 Tecnologías y Herramientas	10
1.5 Conclusiones parciales	17
Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta	19
2.1 Introducción.....	19
2.2 Planificación.....	19
2.3 Equipo y flujo de trabajo.....	19
2.4 Captura de requisitos	20
2.5 Requisitos funcionales del sistema.	21
2.6 Requisitos no funcionales	23
2.6.1 Requisitos del hardware.	23
2.6.2 Requisitos del software.....	24
2.7 Historias de Usuario(UH).	24
2.8 Análisis de los Costos	26
2.9 Plan de Iteraciones	29
2.10 Reuniones	30
2.11 Plan de entregas	30
2.12 Etapa de diseño	31
2.13 Prototipo de interfaz de usuario	32
2.14 Tareas a desarrollar	33
2.14.1 Tarjetas de Clase, Responsabilidad y Colaboración.....	37
2.14.2 Modelo físico de la base de datos.....	40
2.15 Conclusiones parciales	41

ÍNDICE

Capítulo 3: Validación de la solución propuesta.....	42
3.1 Introducción	42
3.2 Pruebas al software	42
3.2.1 Plan de pruebas.....	42
3.2.3 Pruebas Funcionales:	44
3.3 Herramientas de Pruebas.	48
3.4 Análisis de los resultados obtenidos.	48
3.5 Conclusiones Parciales.....	49
Conclusiones Generales	50
Recomendaciones.....	51
Bibliografía	52
Anexos	55

INTRODUCCION

Introducción

En la actualidad la tecnología se ha convertido en parte fundamental de la vida cotidiana, tanto es así, que hasta en las actividades alejadas de esta tecnología se ha convertido en un medio masivo para llevar información a personas en cualquier región del mundo. Un ejemplo de ello da lugar a la aparición de la internet, la red de redes.

Actualmente, la sociedad demanda nuevas y mejores herramientas informáticas para registrar y obtener información de los diversos procesos, en este ámbito las tecnologías de la información y comunicación juegan un papel fundamental, al proporcionar grandes ventajas tanto desde el punto de vista de agilidad y sencillez como del volumen físico de la información resultante del proceso realizado. De ahí que se desarrolle la necesidad de elaborar técnicas y programas computacionales que permitan una mejor organización de la información además de hacer más sencillo el trabajo a realizar. Facilitando así la obtención de programas informáticos eficientes y funcionales.

En la Universidad de Matanzas “Camilo Cienfuegos” se recoge toda la información referente al control de las actividades docentes a través de documentos Word. Este proceso causa un mal desarrollo de las actividades referentes a este tema por no tener una coordinación centralizada, además de la desorganización en cuanto a la información que se recoge en dicho proceso y problemas en la seguridad y transparencia del proceso.

Según entrevistas realizadas al personal que interviene en el proceso de control a las actividades docentes que se desarrollan en la Universidad de Matanzas, se pudo llegar a la conclusión que no se estaba llevando un proceso organizado y seguro, además de que el mismo se hace de forma manual y se guardan en texto plano. Dentro de los problemas que se encuentran en el proceso, es la descoordinación que existe entre los sujetos involucrados y la información que se recoge en el proceso de control por la falta de una aplicación que oriente y organice dicho proceso.

Lo anteriormente planteado permite al autor plantearse como **problema científico**: ¿Cómo automatizar el proceso de control a las actividades docentes de la Universidad de Matanzas?

INTRODUCCION

Para resolver este problema se determinó como **objetivo general**: Desarrollar una aplicación web para aumentar la eficiencia del control al proceso de evaluación de las actividades docentes de la Universidad de Matanzas.

En la actualidad el proceso donde se gestiona la información obtenida del control a las actividades docentes es una tarea bastante engorrosa debido a que todo este volumen de información es tratado de manera manual y se va registrando en papel, haciendo así el trabajo más difícil y desorganizado, de ahí la necesidad de crear un sistema informático que agilice y organice este proceso.

Para darle cumplimiento al objetivo se tienen como objetivos específicos los siguientes:

- 1- Elaborar los fundamentos teóricos metodológicos para el desarrollo del sistema web para para aumentar la eficiencia del control al proceso de evaluación de las actividades docentes.
- 2- Diseñar del sistema web para para aumentar la eficiencia del control al proceso de evaluación de las actividades docentes de la Universidad de Matanzas.
- 3- Implementación de la aplicación web para aumentar la eficiencia del control al proceso de evaluación de las actividades docentes de la Universidad de Matanzas.
- 4- Validación de la aplicación web para aumentar la eficiencia del control al proceso de evaluación de las actividades docentes de la Universidad de Matanzas.

Una vez llevadas a cabo eficientemente las tareas de investigación propuestas con anterioridad, se espera lograr un software capaz de:

- 1- Gestionar controles de las actividades docentes.
- 2- Elaborar dinámicamente los parámetros que se evaluarán en el control a las actividades docentes.
- 3- Sugerir una evaluación para los controles.
- 4- Gestionar una evolución cuantitativa de los parámetros a evaluar los controles y generar una gráfica para ver el comportamiento y desempeño de los profesores en las actividades docentes.
- 5- Generar reportes.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Efecto económico

Este proyecto representa una mejora significativa en el proceso docente de la Universidad de Matanzas al poder gestionar todo el proceso de control a las actividades docentes de manera automatizada. Por ser un software que no necesitara prácticamente de mantenimiento debido a que esta creado lo más dinámicamente posible, es decir que en futuros cambios de las resoluciones donde se vea afectado este proceso de control, este software se puede adaptar a los cambios, evitando así los problemas de mantenimientos como el excesivo costo de los mismos. Esta herramienta informática está orientada también en ahorrar el tiempo y esfuerzo de todo el personal que interviene en el proceso de control a las actividades docentes, así como organizar y coordinar toda la información referente al mismo.

Estructura de la investigación

La investigación se dividirá en 4 capítulos, los cuales se describirán a continuación:

Capítulo I. Marco Teórico Referencial

En el marco teórico referencial se describen los procesos relacionados con el control de las actividades de la universidad a partir del análisis de los documentos rectores. Posteriormente se determina la metodología, las herramientas y la arquitectura para el desarrollo de la aplicación propuesta.

Capítulo 2. Análisis, diseño y desarrollo de la solución propuesta

Presenta una propuesta de solución para el sistema, donde se describen las reglas y los elementos del negocio, una planificación inicial del proyecto con el empleo de la metodología de Ingeniería del Software *eXtreme Programming* (XP). La solución que se propone se basa en el análisis y diseño del sistema; tomando en cuenta los intereses originados por el cliente, los cuales se describen mediante las Historias de Usuario.

Capítulo 3. Casos de Prueba

Se le realizarán una serie de pruebas al sistema para medir su funcionamiento y la calidad del mismo.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Capítulo 1: Marco teórico referencial

1.1 Introducción

Este capítulo contiene los conceptos fundamentales relacionados con el objeto de estudio, los antecedentes del trabajo, los métodos de la investigación. Aborda las tecnologías, herramientas y metodologías que se utilizarán durante el desarrollo de la investigación.

1.2 Objeto de Estudio

El objeto de estudio de esta investigación se enmarca en la gestión del control a las actividades docentes de la Universidad de Matanzas, por tanto, se hace necesario hacer una descripción más exacta de este elemento que se pretende transformar y perfeccionar.

1.2.1 Caracterización del proceso de Control a las Actividades Docentes.

El proceso de control a las actividades docentes es uno de los procesos más importantes en el ámbito universitario debido a que retroalimenta a los ejecutores y a los dirigentes del estado real de cada uno de ellos. En las universidades las actividades docentes juegan un papel esencial pues son aquellas en las cuales se organiza o se materializan los procesos formativos. De ahí que el control a estas actividades sea de vital importancia para la comunidad universitaria pues impacta directamente en la formación del profesional. Para la vicerrectoría docente de la Universidad de Matanzas no todas las actividades docentes tienen el mismo peso ni se evalúan de la misma manera, aunque reconoce la significativa importancia de la actividad denominada clase. Esta actividad se ejecuta a partir de la carrera y la disciplina para detectar los errores que se cometen lo cual permite la adopción de estrategias metodológicas para su corrección. En estos controles pueden intervenir varios controladores y controlados, en dependencia del nivel y el tipo de actividad a controlar. Cada uno de estos controles genera la documentación necesaria para describir la evaluación obtenida y su calificación en dependencia de la norma que haya establecido la institución. A partir de las entrevistas realizadas a varios actores de este proceso como son vicerrectora docente, dos vicedecanos docentes, tres jefes de carreras y el análisis documental se obtuvieron los siguientes gráficos:

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

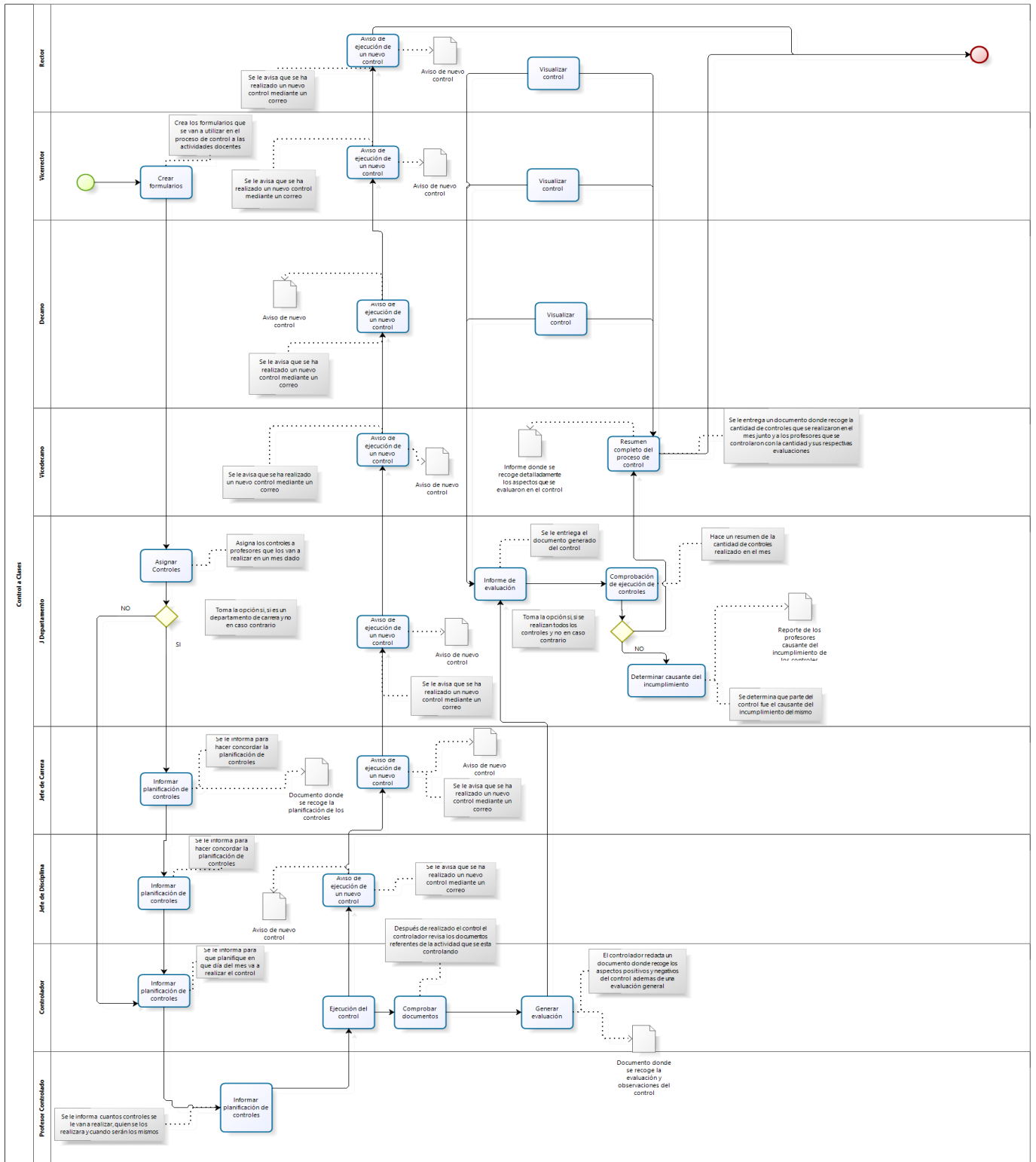


Figura # Mapa de procesos de ///. Fuente: Elaboración del autor
Diagrama de proceso del negocio (Orientado al control a clases)

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

En este acápite se describirá de manera general el proceso de control de actividades. El control a clases como se observa en la figura comienza con el vicerrector creando el formulario que se va tomar como guía en dicho proceso, siguiendo el flujo el jefe de departamento asignaría los controles que les tocaría a cada profesor y qué papel va a desempeñar en el mismo, si es un departamento de carrera se le informa al jefe de carrera y al de disciplina para una mejor organización sino se les notifica directamente a los profesores involucrados en los controles. A continuación, se realiza el control que comienza comprobando la actividad y posteriormente se solicitan los documentos. Este cruzamiento se realiza con el objetivo de contrastar lo ejecutado con lo planificado.

Para evaluar el control a las actividades docentes es importante comprender que no todas las actividades poseen la misma importancia. La actividad docente de mayor importancia es la clase, sin embargo, sus parámetros de evaluación varían en dependencia de los objetivos de cada año. Sucede de manera análoga para el resto de los controles. Ello hace que se tome en cuenta para la evaluación de las actividades se asuman un conjunto de factores, atributos y criterios de medidas. Para tener un criterio cualitativo de estos procesos se construye la siguiente métrica:

$$v = \sum_{f=1}^F P_f \sum_{a=1}^A P_a \sum_{cm=1}^{CM} Cm_{cm} * P_{cm}$$

Ecuación #1: Cálculo del valor de evaluación de actividad. Elaboración del autor.

Esta métrica permite el establecimiento de escalas nominales definida por el rol que establece los valores que pueden tomar cada una de las variables que intervienen.

1.3 Métodos de la investigación

1.3.1 Métodos teóricos empleados

- **Análisis Histórico-lógico:** Permite esclarecer las etapas de desarrollo histórico del problema a solucionar y cómo funciona la gestión de los proyectos de investigación, su definición, características, evolución y desarrollo.
- **Hipotético-deductivo:** Desempeñó un papel esencial en el proceso de verificación del supuesto de partida. A partir de este supuesto se pueden inferir conclusiones y establecer predicciones.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- **Analítico-sintético:** 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.
- **Enfoque de sistema:** Proporcionó la orientación general del estudio como una realidad integral formada por elementos que interaccionan unos con otros.

1.3.2 Métodos empíricos empleados

- **Observación:** Acompañó la investigación desde los primeros momentos, a través de la cual se conoció y se observó cómo se lleva a cabo el proceso de gestión de los proyectos de investigación y las dificultades que ello conlleva, lo que condujo a la determinación de los procesos que se automatizan a partir de la implementación de la aplicación web.
- **Entrevista:** Permitió conocer cuáles eran las inquietudes y sugerencias por parte del personal especializado en el tema y qué hacer para facilitar y agilizar su trabajo.
- **Análisis de documentos:** Incluye la revisión de documentos como son las resoluciones, decretos y leyes vigentes relacionados con el tema. También se consultaron libros y diversos artículos de bancos de información de Internet relacionados con el tema a investigar.

1.4 Herramientas, tecnologías y metodologías de desarrollo

Para el desarrollo de esta investigación, se hace necesario el estudio y selección de herramientas, tecnologías y metodologías de desarrollo con el propósito de darle cumplimiento al objetivo general. A continuación, se describen aspectos a tener en cuenta.

1.4.1 Tendencias tecnológicas a considerar

Arquitectura del software

La arquitectura de software proporciona las bases para el desarrollo e implementación de sistemas informáticos. En el caso de las aplicaciones web desarrolladas en la actualidad se debe tener en cuenta la arquitectura Cliente/Servidor y el patrón de arquitectura Modelo Vista Controlador (MVC), ambos elementos se describen a continuación.

Arquitectura Cliente-Servidor

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. El servidor es la integración de una máquina con el hardware necesario para responder las peticiones de los clientes con el software necesario para ello. Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Esta arquitectura también se puede aplicar a programas que se ejecutan sobre un solo ordenador, aunque es más conveniente utilizarla en un sistema operativo multiusuario distribuido a través de una red de computadoras. De aquí la ventaja de realizar una aplicación web, porque una vez instalado el software en un servidor, este podrá ser accedido por los usuarios de la aplicación a través de una red, mediante un cliente en una terminal.

Ventajas del Esquema Cliente/Servidor:

Administración centrada en el servidor: Los clientes tienen poca trascendencia en el esquema y sus necesidades de administración son menores.

Centralización de los recursos: Los recursos comunes a todos los usuarios se administran en el servidor. Así se evitan situaciones como la redundancia o inconsistencia de información en las bases de datos.

Mejora de la seguridad: Al disponer de un mecanismo central de autenticación, las posibilidades de acceso indebido se reducen considerablemente.

Escalabilidad de la instalación: Se pueden añadir o suprimir clientes sin que el funcionamiento de la red se vea afectado.

Una representación gráfica de este tipo de arquitectura sería la siguiente:

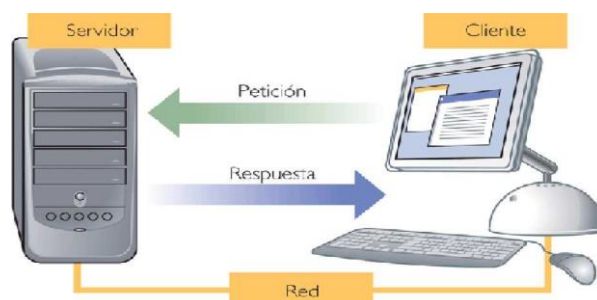


Figura1.1: Tomada de clases

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios con la respuesta (provee el servicio), dicha arquitectura permite distribuir físicamente los procesos datos en la forma más eficiente. Esta idea se puede

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

aplicar a programas que se ejecutan sobre una misma computadora puesto que es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Patrones de diseño:

Los patrones de diseño son un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Estos identifican y especifican abstracciones que van más allá del simple ámbito de clases e instancias, o componentes. Para que una solución sea considerada un patrón debe poseer ciertas características como son efectividad habiendo resuelto problemas similares en ocasiones anteriores y reusabilidad permitiendo su aplicación a diferentes problemas de diseño en distintas circunstancias (Arrakis 2007).

Se decide utilizar el patrón de diseño Modelo-Vista-Controlador, el cual se explica a continuación:

Modelo-vista-controlador (MVC)

El modelo-vista-controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (García 2016).

De manera genérica, los componentes de MVC se podrían definir como sigue:

El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tantas consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). El modelo envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). Se podría decir que el controlador hace de intermediario entre la vista y el modelo.

La **Vista**: Presenta el modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho modelo la información que debe representar como salida.

Software libre

El término *software* libre se refiere a un programa de ordenador con libertad para su utilización, distribución, modificación y estudio. Desde el punto de vista técnico-legal, se considera *software* libre a los programas licenciados en términos que garantizan a sus usuarios el derecho de ejecutarlos, copiarlos, distribuirlos, estudiarlos, cambiarlos y mejorarlos (Tedeschi 2013).

1.4.2 Metodologías de desarrollo

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. Estas 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. La constante innovación tecnológica hace que cada vez sea más necesaria la aplicación de nuevas metodologías adaptadas a los nuevos tiempos (Zanotti 2016).

Actualmente existen dos grupos de metodologías de desarrollo, las **metodologías tradicionales** y las **metodologías ágiles**. Las primeras se centran en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto, mientras que las segundas dan mayor importancia a la capacidad de respuesta a los cambios. A continuación, se presenta una breve comparación entre ellas.

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.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

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. Entre las metodologías ágiles más conocidas se encuentran: *Scrum*, *Crystal Metodologies* y *Extreme Programming (XP)*.

Para el desarrollo de esta investigación, se hace necesario el estudio y selección de herramientas, tecnologías y metodologías de desarrollo con el propósito de darle cumplimiento al objetivo general. A continuación, se describen aspectos a tener en cuenta.

Metodología de desarrollo XP (Programación Extrema)

Es una metodología ágil que se centra principalmente en las necesidades del cliente y en las relaciones interpersonales como clave para el éxito en el desarrollo del software. Se basa en el trabajo en equipo, siempre preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo en el equipo. Uno de sus principios está en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación constante entre todos los participantes, claridad en las soluciones implementadas y coraje para enfrentar los cambios que se puedan producir en el avance del software (RocketTheme 2009).

Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software (RocketTheme 2009).

Esta metodología de desarrollo de software es una de las más exitosas utilizada en la actualidad para proyectos de corto plazo y con poco personal. Consiste en una programación rápida o extrema. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Está vinculada a cuatro frases, la planificación, el diseño, la codificación y las pruebas.

Algunas de las características con las que cuenta esta metodología son:

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Es una metodología ágil pensada para proyectos cortos con requerimientos muy cambiantes y pequeños grupos de desarrollo.
- Es una metodología orientada fuertemente hacia la codificación.
- Es un proyecto pequeño donde todo el trabajo se realiza por una pareja de programadores.
- La generación de numerosos artefactos y roles no es necesaria debido a que el proyecto es a corto plazo y está centrado en ser desarrollado en el menor tiempo posible.
- Su idea es ofrecer entregas del sistema agregando funcionalidades paulatinamente en corto tiempo. Esto permite que el usuario tenga más claro si el sistema hace lo que él quiere y lo compromete en el desarrollo.
- Presenta un diseño evolutivo, lo cual hace que no se le de mucha importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

Se decidió utilizar XP debido a que se adapta en gran medida tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo. Ya que el proyecto es pequeño y XP está concebida para ser utilizada dentro de proyectos pequeños, no existe un contrato previo especificando tiempo, recursos y alcance. Para el desarrollo del sistema no se dispone de un contrato con un presupuesto ni un alcance previamente definidos. El cliente forma parte del equipo de desarrollo, mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades, el riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado.

XP está diseñada a mitigar los riesgos en proyectos con estas características y existe poca disponibilidad de personal. El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.

Justificación de su uso en el proyecto

Para el desarrollo de este proyecto se ha decidido utilizar como metodología de desarrollo XP para darle solución a la propuesta, pues, por estar dentro del grupo

de metodologías ágiles, se caracteriza por su sencillez tanto en su aprendizaje, como en su aplicación. Es adecuada para cualquier tipo de proyecto y para equipos de cualquier tamaño. XP es de todas las metodologías ágiles citadas, con la que más se ha trabajado y con la que más experiencia se ha adquirido por parte del autor de este trabajo.

1.4.3 Tecnologías y Herramientas

Lenguajes de programación web: En la actualidad existen distintos lenguajes de programación para la Web, principalmente se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente - Servidor. El primer grupo está formado por los lenguajes que se ejecutan en el **lado cliente:** HTML, JavaScript y CSS, todos encargados de aportar dinamismo a la aplicación en los navegadores. También existe un segundo grupo de lenguajes que se ejecutan en el lado servidor entre ellos se encuentran PHP, Java y Python, estos se caracterizan por desarrollar la lógica de negocio, además de ser los encargados del acceso a Bases de Datos y el tratamiento de la información. Las características de los lenguajes empleados se muestran a continuación.

Del lado del cliente:

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, en el proyecto se emplearon las siguientes tecnologías:

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, entre otras), 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. (Amaro Calderón 2007)

CSS (*Cascading Style Sheets*)

Las hojas de estilo en cascada son un mecanismo que permite aplicar formato a los documentos escritos en HTML (y en otros lenguajes estructurados, como

XML) separando el contenido de las páginas de su apariencia. Para el diseñador, esto significa que la información estará contenida en la página HTML, pero este archivo no debe definir cómo será visualizada esa información. Las indicaciones acerca de la composición visual del documento estarán especificadas en el archivo de la CSS. Lo que posibilita crear páginas web de una manera más exacta (Álvarez 2012)

JavaScript

Se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. 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. A pesar de su nombre, legalmente *JavaScript* es una marca registrada de la empresa Sun Microsystems (Cristian 2008)

Del lado del servidor:

PHP

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. PHP es un acrónimo de “Hypertext Preprocessor”, es un lenguaje “Open Source” interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP (Pérez 2009).

Justificación para su uso en el proyecto

El autor determinó para el desarrollo de la solución propuesta, el lenguaje de programación PHP porque es multiplataforma y debido sus características se hace posible que el cliente interactúe con una página rápida, eficiente y segura; capaz de mostrar y procesar información de forma eficiente.

Unido a lo anterior, se puede mencionar que es muy sencillo y legible, y cuenta con una amplia librería de funciones, que permiten hacer cualquier tipo de operación, como trabajo con archivos y carpetas, procesamiento de formularios,

paginación de resultados y muchas otras opciones. Esta amplia librería concede la posibilidad de realizar un producto de acuerdo a las necesidades del usuario. Además, PHP es, de todos los Lenguajes de Programación estudiados en la carrera, con uno de los que más se ha trabajado y experiencia se ha adquirido.

Frameworks: Un *framework* constituye una estructura de soporte definida (o plataforma de desarrollo enfocada al entorno web) en la cual otro proyecto de software puede ser organizado y desarrollado (Achour 2007).

Symfony

Es un *framework* PHP que permite utilizar fácilmente la arquitectura MVC (Model-View-Controller). Fue diseñado para optimizar el desarrollo de aplicaciones web, proporcionando herramientas para agilizar aplicaciones complejas y guiando al desarrollador a acostumbrarse al orden y buenas prácticas dentro del proyecto. *Symfony* reutiliza conceptos y desarrollos exitosos de terceros y los integra como librerías para ser utilizados por nosotros. Entre ellos encontramos que integra plenamente uno de los *frameworks* ORM más importantes dentro de los existentes para PHP llamado **Doctrine**, el cual es el encargado de la comunicación con la base de datos, permitiendo un control casi total de los datos sin importar si es un SGBD como MySQL, PostgreSQL, SQL server, Oracle, entre otros motores. Otro ejemplo de esto es la inclusión del framework **Twig**, un poderoso motor de plantillas que nos permite separar el código PHP del HTML permitiendo una amplia gama de posibilidades y por sobre todo un extraordinario orden para el proyecto (Alcalde 2013).

Justificación para su uso en el proyecto

Symfony se seleccionó porque se ajusta a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del Sistema Gestor de Bases de Datos (SGBD). Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, está más indicado para grandes aplicaciones Web que para pequeños proyectos.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Está basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- 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.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código ahorrándonos tiempo de trabajo.

Bootstrap

Bootstrap es un framework que permite montar una estructura responsive fácilmente. Se plantea la facilidad porque muchas clases y funciones están desarrolladas, incorpora muchas librerías como normalice, jQuery, less, por lo que se puede hacer uso de muchos efectos y funciones sin tener que programar de cero. Permite, además, conseguir un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones (Bowler and Bancner 2009).

Características principales de Bootstrap

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
- Incorpora un sistema de grid basado en 12 columnas fluidas para poder acoplarse en función del dispositivo desde el que se acceda.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo JQuery.
- Es un framework ligero que se integra de forma limpia con el proyecto actual.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Funciona con todos los navegadores, incluido Internet Explorer usando HTMLShim para que reconozca los tags HTML5.

jQuery

Es un framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX, además de ser software libre y de código abierto. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Otra gran ventaja de JQuery es que permite cambiar el contenido de la página web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis (Risueño 2013).

Sistemas de Gestión de Base de Datos (SGBD): los Sistemas Gestores de Bases de Datos (SGBD, en inglés DBMS: *DataBase Management System*) son un tipo de software que permiten 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 (Chaffer 2009).

MySQL

Es un gestor de base de datos sencillo de usar e increíblemente 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. 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. Es gratis para aplicaciones no comerciales. Dentro de las principales características se encuentra, la gran portabilidad entre sistemas, soporta gran cantidad de tipos de datos para las columnas y hasta 32 índices por tabla, manteniendo un buen nivel de seguridad pues gestiona los datos de los usuarios y contraseñas, dispone de API's en gran cantidad de lenguajes (Python, C, C++, Java, PHP) y aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multi-hilo (Marcano 2009).

Justificación para su uso en el proyecto

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Se eligió MySQL, porque es uno de los servidores de bases de datos de código abierto más populares y conocidos del mundo. Posee un sistema de manejo de bases de datos con un gran nivel de estabilidad y facilidad de desarrollo que se integra fácilmente con el lenguaje de programación PHP. Dispone, además, de una arquitectura que lo hace extremadamente rápido y fácil de personalizar. Sumándole a todos estos beneficios, es importante considerarlo como un servidor que se adecúa perfectamente a las exigencias del cliente.

Servidor web: Apache

El servidor web HTTP seleccionado para la aplicación es Apache, es usado principalmente para enviar páginas web estáticas y dinámicas en la WWW, así como PHP y MySQL, también es liberado bajo licencia Open Source y es multiplataforma entre ellas las más populares como Unix, Microsoft Windows y Macintosh.

El servidor **HTTP Apache** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual (Martínez 2014). Apache es el servidor web por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Apache soporta servir diversos sitios web con un sólo servidor. Para ello proporciona facilidades de creación de dominios virtuales en función de diversas direcciones IP o diversos nombres por IP (Morine 2013).

Algunas de las características que contribuyen con la popularidad de este servidor son:

- Se ejecuta en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierta.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.

También dispone de una amplia variedad de módulos, que permiten generar contenido dinámico (con PHP, Java, Perl, Python y otras), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, crear servidores virtuales por IP o por nombre (varias direcciones Web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

módulos están disponibles junto con su código fuente, por lo cual pueden ser incluso hasta modificados por cualquier persona con conocimientos de programación. Teniendo en cuenta las características antes mencionadas se escoge el servidor Web Apache como servidor Web para la aplicación que será desarrollada.

Doctrine.

Es un asignador objeto relacional (ORM) para PHP 5.3.0+ que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una capa de abstracción de base de datos (DBAL por Data Base Abstraction Layer). La principal tarea de los asignadores de objetos relacionales es la traducción transparente entre objetos (PHP) y las filas relacionales de la base de datos. Una de las características clave de Doctrine es la opción de escribir las consultas de base de datos en un dialecto SQL propio orientado a objetos llamado Lenguaje de Consulta Doctrine (DQL por Doctrine Query Language), inspirado en Hibernate HQL. Además, DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible

Entorno de desarrollo (IDE): Es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como PHP, C++, Java, C#, Delphi, Visual Basic, Python, entre otras.

PhpStorm

Es un potente entorno de desarrollo integrado (IDE), especialmente diseñado a fin de proporcionar a los desarrolladores de HTML, JavaScript y PHP todas las herramientas necesarias para su trabajo. PhpStorm proporciona un editor de código enriquecido e inteligente para PHP con resaltado de sintaxis, configuración extendida de formateo del código, navegación rápida y comprobación de errores sobre la marcha y finalización de código inteligente (Mateu 2004).

AJAX (Asynchronous JavaScript and XML)

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Según lo planteado porAJAX no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen para lograr funcionalidades interesantes en la web. AJAX, en resumen, es el acrónimo para Asynchronous JavaScript + XML y el concepto es: cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo renderizando la página y mostrando u ocultando porciones de la misma. De esta forma, es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. No tiene nada que ver con un lenguaje de programación, quizás se parezca más al uso de un procesador de texto por la utilización de códigos de comienzo y finalizado de estilo. Además de resultar más sencillo, no se necesita ninguna herramienta de programación, compiladora o similar, sino que cualquier editor de texto puede servir para realizar las páginas web.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor. Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

1.5 Conclusiones parciales

Con la elaboración de este capítulo quedaron establecidas las bases teóricas que sustentan esta investigación, a partir del estudio de los antecedentes y tendencias actuales del software para la gestión de los Proyectos de Investigación en el departamento de Ciencia, Técnica e Innovación de la universidad de Matanzas y se ha llegado a las siguientes conclusiones:

- Se especificaron los elementos esenciales del proceso de gestión de los proyectos de investigación de la Universidad de Matanzas.

CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Se determinó utilizar Symfony como *framework*, PHP como lenguaje de programación, MySQL como sistema gestor de base de datos, Xamp como servidor web, XP como metodología de desarrollo de software, MVC como patrón de diseño y la arquitectura cliente-servidor.

Capítulo 2: Análisis, diseño y desarrollo de la solución propuesta

propuesta

2.1 Introducción.

En este capítulo, apoyándose en la metodología de desarrollo de software Programación Extrema (XP), se abordan los elementos necesarios para describir la solución propuesta. Debido a la metodología se han usado las Historias de Usuarios para el levantamiento de requisitos, con la cual también se realiza la planificación inicial del proyecto y un estudio de factibilidad con el análisis de costo y beneficio para determinar si es factible el desarrollo del sistema. Esta metodología se decidió utilizar debido a que se basa en la comunicación, simplicidad, retroalimentación y reutiliza el código de desarrollo y su principal objetivo es satisfacer al cliente.

2.2 Planificación.

Es importante para tener una solución del problema una planificación, desde el inicio pensar en soluciones y organizar el modelado inicial del problema, tener un enfoque real de la problemática a que enfrentamos y así una solución efectiva. Se empleó las historias de usuarios para modelar y organizar las ideas del cliente, obteniendo un punto de partida para el resto de la planificación del proyecto. Es necesario estimar el tiempo y una planificación inicial de las entregas, hay que tener en cuenta que pueden surgir nuevos requisitos, esto puede afectar esa planificación inicial.

2.3 Equipo y flujo de trabajo

La base del éxito del desarrollo de un proyecto está en gran medida en el valor del **equipo de trabajo** que lo lleva a cabo. Los integrantes del equipo que realizaron esta investigación fueron:

Miembros	Roles
DrC. Walfredo González	Jefe de Proyecto, Tester
Dra. C. Leticia Fuentes	Cliente
Wuilliam Hernández Rodríguez	Programador, Tester

Tabla 1: Equipo de trabajo y roles

Para un mejor entendimiento del equipo de trabajo es importante ver como se aplican las prácticas según la metodología XP:

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

- La presencia y disponibilidad del cliente es importante dentro del equipo, las relaciones con este son muy necesarias pues las personas vinculadas de forma directa con el desarrollo del sistema en caso de cualquier duda pueden acceder a él rápidamente, evitando atrasos innecesarios durante el proceso de desarrollo.
- El diseño debe ser simple y los programadores han de seguir un estándar de programación o deben tener el mismo estilo programando. El código debe ser reestructurado contantemente con el objetivo de remover duplicaciones, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- Se deben ir programando pequeñas versiones que van aumentando la dimensión poco a poco y en caso de existir fallos se lleva a cabo la reprogramación del código sin variar su funcionalidad, cada pieza de código es integrada en el sistema una vez que esté lista, así el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- Se deben ir proponiendo pequeñas versiones que irán aumentando en complejidad y dimensión en la medida del tiempo y en caso de existir alguna anomalía se lleva a cabo la reprogramación del código sin modificar su principio funcional. Cada segmento de código es integrado en el sistema una vez que esté listo, de esa forma el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- El tiempo de trabajo por semana no debe exceder las 40 horas, esto contribuye a evitar el cansancio mental de los desarrolladores y con ello disminuye la probabilidad de errores.
- La producción de código está dirigida por las pruebas unitarias las cuales tienen como objetivo principal evitar errores y que estos se detecten lo más rápido posible. Las pruebas se convierten en una herramienta de desarrollo, no un paso de verificación que puede despreciarse si a uno le parece que el código está bien.

2.4 Captura de requisitos

La captura de requisitos es la disciplina mediante la cual se recopila la información y se transforma en un conjunto de requerimientos que son los que darán límite al alcance del sistema.

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

El desarrollo de esta disciplina en el equipo de trabajo fue posible gracias al estudio bibliográfico de los antecedentes, mediante sistemas de información y el intercambio con los expertos, quienes aportaron una importante guía en la formalización de esta base de conocimientos. La entrevista fue la técnica de mayor valor en la obtención de datos relevantes acerca de las necesidades reales del servicio, el establecimiento de un vocabulario común, la identificación de procesos, la recopilación y consenso de requisitos.

2.5 Requisitos funcionales del sistema.

Un requerimiento es una característica que el sistema debe tener y, a la vez, es una restricción que el sistema debe cumplir para ser aceptada por el cliente. El levantamiento de requisitos es una parte fundamental dentro del proceso de creación del software, pues rige la línea de desarrollo, por ser el que registra las características y condiciones que debe tener el sistema final.

Para que el sistema se desarrolle con las condiciones y calidad necesarias el cliente ha requerido que se cumplan los siguientes requisitos:

- Autenticarse.
- Gestionar usuarios.
 - Insertar usuario.
 - Editar usuario.
 - Eliminar usuario.
 - Listar usuario.
 - Asignar roles
- Gestionar Categoría Investigativa.
 - Insertar Categoría Investigativa.
 - Editar Categoría Investigativa.
 - Eliminar Categoría Investigativa.
 - Listar Categoría Investigativa.
- Gestionar Categoría Profesional.
 - Insertar Categoría Profesional.
 - Editar Categoría Profesional.
 - Eliminar Categoría Profesional.
 - Listar Categoría Profesional.

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

- Gestionar Categoría Docente.
 - Insertar Categoría Docente.
 - Editar Categoría Docente.
 - Eliminar Categoría Docente.
 - Listar Categoría Docente.
- Gestionar Especialidad.
 - Insertar Especialidad.
 - Editar Especialidad.
 - Eliminar Especialidad.
 - Listar Especialidad.
- Gestionar Profesor.
 - Insertar Profesor.
 - Editar Profesor.
 - Eliminar Profesor.
 - Listar Profesor.
- Gestionar Actividades Metodológicas.
 - Insertar Actividades Metodológicas.
 - Editar Actividades Metodológicas.
 - Eliminar Actividades Metodológicas.
 - Listar Actividades Metodológicas.
- Gestionar Facultad.
 - Insertar Facultad.
 - Editar Facultad.
 - Eliminar Facultad.
 - Listar Facultad.
- Gestionar Departamento.
 - Insertar Departamento.
 - Editar Departamento.
 - Eliminar Departamento.
 - Listar Departamento.
- Gestionar Carrera.
 - Insertar Carrera.

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

- Editar Carrera.
- Eliminar Carrera.
- Listar Carrera.
- Gestionar Grupo.
 - Insertar Grupo.
 - Editar Grupo.
 - Eliminar Grupo.
 - Listar Grupo.
- Gestionar Asignatura.
 - Insertar Asignatura.
 - Editar Asignatura.
 - Eliminar Asignatura.
 - Listar Asignatura.
- Gestionar Tipo de Clase.
 - Insertar Tipo de Clase.
 - Editar Tipo de Clase.
 - Eliminar Tipo de Clase.
 - Listar Tipo de Clase.
- Gestionar Evaluación del Control.
 - Insertar Evaluación del Control.
 - Editar Evaluación del Control.
 - Eliminar Evaluación del Control.
 - Listar Evaluación del Control.

- Generar Reportes de Evaluación del Control.
 - Imprimir reporte de un control con respecto a una evaluación.

2.6 Requisitos no funcionales

Son aquellos requisitos que se encargan de especificar características como aquella que necesita el software para su funcionamiento, costo, usabilidad y rasgos de manera general establecidos por el cliente.

2.6.1 Requisitos del hardware.

Servidor:

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

- Procesador Pentium III o superior, 256MB de memoria RAM como mínimo.

Cliente:

- Display con resolución 1366 x 768.
- Procesador Pentium III o superior, 256 MB de RAM como mínimo.
- Windows NT o superior.
- Conexión de red.

2.6.2 Requisitos del software.

- Servidor con IIS, que soporte PHP, o con Apache.
- Servidor de Base de Datos: MYSQL.
- Navegador de Internet que soporte HTML5, preferentemente Google Chrome con el plugin NoScript para evitar fallas de seguridad.

2.7 Historias de Usuario(UH).

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario (UH User History en inglés) con el cliente. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una UH, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Escala Nominal de Prioridad

Alta: Se le otorga a las Historias de Usuario que resultan funcionalidades fundamentales en el desarrollo del proyecto, a las que el cliente define como principales para el control integral de proyectos.

Media: Se le otorga 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 son esenciales para el proyecto en desarrollo.

Escala Nominal de Riesgo en Desarrollo (Pressman, 2010)

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

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

Media: Cuando pueden aparecer errores en la implementación de la UH 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.

En la siguiente tabla se muestran un resumen de todas las Historias de Usuarios (HU) planificadas inicialmente, en las que queda definido el nivel de prioridad con el que deben darle solución a las HU, teniendo en cuenta el riesgo en desarrollo. Se define también la estimación del esfuerzo requerido (en semanas), que no es más que el tiempo en el que se concibió inicialmente el desarrollo de cada HU.

No.	Historia de Usuario	Prioridad	Riesgo	Esfuerzo	Iteración	Entrega
1	Diseño y creación de la Base de Datos.	Alta	Alto	1.5	1	1
2	Diseño de la interfaz de usuario.	Media	Bajo	1	1	1
3	Autenticarse.	Alta	Medio	0.5	1	1
4	Gestionar usuarios.	Media	Medio	1	2	2
5	Gestionar Categoría Investigativa.	Media	Bajo	0.5	2	2
6	Gestionar Categoría Profesional.	Media	Bajo	0.5	2	2
7	Gestionar Categoría Docente.	Media	Bajo	0.5	2	2
8	Gestionar Especialidad.	Media	Bajo	0.5	2	2
9	Gestionar Profesor.	Alta	Medio	1	3	3

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

10	Gestionar Actividades Metodológicas.	Alta	Alto	3	3	4
11	Gestionar Facultad.	Media	Bajo	0.5	4	5
12	Gestionar Departamento.	Media	Bajo	0.5	4	5
13	Gestionar Carrera.	Media	Bajo	0.5	4	5
14	Gestionar Grupo.	Media	Bajo	0.5	4	5
15	Gestionar Asignatura.	Media	Bajo	0.5	4	5
16	Gestionar Tipo de Clase.	Media	Bajo	0.5	4	5
17	Gestionar Evaluación del Control.	Alta	Alto	3	5	6
18	Generar Reportes de Evaluación del Control.	Media	Bajo	1	6	7
19	Gestionar Trazas	Baja	Bajo	1	6	7
Totales				18	7	7

Tabla 2: Historias de usuarios iniciales Fuente: Elaboración del autor

2.8 Análisis de los Costos

La estimación es el proceso de medición anticipada de la duración, esfuerzos y costes necesarios para realizar todas las actividades y obtener todos los productos asociados a un proyecto.

Es necesario tener en cuenta numerosos aspectos que afectan a la estimación como la complejidad del proyecto, su estructuración, el tamaño, los recursos involucrados y los riesgos asociado (Fernández 2011). La estimación del costo de un software es el proceso de predecir la cantidad de esfuerzo requerido para

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

el desarrollo del sistema y el tiempo para ello. Existen diversos modelos para realizar la estimación del costo de un software como, por ejemplo: COCOMO I, COCOMO II, Puntos de Función, Botton – Up, Top – Down, entre otros.

Puntos de Función

Es un modelo 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. Es de especial importancia cuando se necesitan realizar estimaciones tempranas del proyecto como es el caso de esta aplicación.

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.

Una vez obtenidos los diferentes elementos del sistema se utilizan las tablas reflejadas en el **Anexo 1** para asignar pesos en función del número de atributos que tengan y el número de archivos a los que afecte.

La tabla que se muestra a continuación contiene el resultado de los componentes obtenidos por su peso.

Componentes	Cantidad de Componentes por su Peso	Total
EI (Entradas externas)	28 * 3	84
EO (Salidas externas)	1 * 4	4
EQ (Consultas)	15* 3	45
ILF (Ficheros Lógicos Internos)	13 * 10	130
ILF (Ficheros Lógicos Externos)	0	0

Tabla---: Resultado de los Componentes por su peso

Cálculo de los Puntos de Función sin Ajustar (PFSA)

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

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

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

$$PFSA = 84 + 4 + 45 + 130 + 0 = 263$$

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 como se muestra en el **Anexo 2**. Cuando cada Factor tenga un valor, se suman todos y así obtenemos el ACT. A continuación, se muestra en una tabla como se refleja dicho procedimiento.

No.de Factor	Factor de Ajuste	0<valor<5
1	Comunicación de Datos	4
2	Proceso Distribuido	3
3	Objetivos de Rendimiento	4
4	Configuración de Explotación Compartida	3
5	Tasa de transacciones	2
6	Entrada de Datos en Línea	4
7	Eficiencia con el Usuario Final	4
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	4
13	Instalaciones Múltiples	4
14	Facilidad de Cambios	2
Total	Ajuste de Complejidad Técnica(ACT)	42

Tabla 3: Obtención del ACT

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

$$PFA = 263 * [0.65 + (0.01 * 42)]$$

$$PFA = 263 * 0.98$$

$$PFA = 281.41$$

Cálculo del Esfuerzo

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

Líneas de Código (LC)

$$LC = PFA * (\text{Líneas por PF})$$

Para calcular las Líneas por PF o Líneas por Puntos de Función nos apoyamos en el **Anexo 3**.

$$LC = 281.41 * 20$$

$$LC = 5628.2$$

Esfuerzo en Horas/Personas (E)

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

$$E = 281.41 / 0.125$$

$$E = 2252 \text{ horas/personas}$$

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

$$2544 \text{ horas/personas} / 1 \text{ persona} = 2544 \text{ horas}$$

$DM = 2252 \text{ horas} / 192 \text{ horas} = 11.72$ meses aproximadamente 12 meses por el margen de error de la estimación por puntos de función.

Costo total del proyecto

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

$$CT = 250 * 1 * 12$$

$$CT = 3000$$

2.9 Plan de Iteraciones

Teniendo las historias de usuario del sistema declaradas y la estimación del esfuerzo para implementar cada una de ellas se procede a realizar la planificación de la etapa de implementación del proyecto. Atendiendo a lo mencionado con anterioridad se decide realizar cinco iteraciones durante el desarrollo del proyecto. A continuación, se muestra el plan de iteraciones:



Figura 1: Planificación de las Iteraciones

2.10 Reuniones

El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo. A pesar de no llevar a cabo una documentación extensa, es muy estricta a la hora de la organización del trabajo. Quincenalmente se realizará una reunión general con el equipo de trabajo con el objetivo de ver el progreso de cada iteración y en caso de algún inconveniente se deberá planificar una reunión extra que deberá ser dentro de dicha quincena.

2.11 Plan de entregas

Las entregas se realizan al terminar cada iteración para tener una mejor comunicación con el cliente, permitir un desarrollo del sistema más eficiente y garantizar que se cometan menos errores.

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

2.12 Etapa de diseño

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.

Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto.

Especificación de las historias de usuario:

En la **Tabla 3** se muestra la HU 1: Diseño y Creación de la base de datos.

Historia de Usuario	
Número: 1	Nombre Historia: Diseño y creación de la Base de Datos
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1.5	Iteración Asignada: 1
Programador responsable: Wuilliam Hernández Rodríguez	
Descripción: Se diseña e implementa la base de datos en el gestor MySQL.	
Observaciones: Verificar la integridad de los datos	

Tabla 4: HU 1: Diseño y Creación de la base de datos

En la **Tabla 12** se muestra la HU 10: Gestionar Actividades Metodológicas

Historia de Usuario	
Número: 10	Nombre Historia: Gestionar Actividades Metodológicas
Usuario: Vicerrector	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 3
Programador responsable: Wuilliam Hernández Rodríguez	

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

Descripción: Mostrar, eliminar, editar y agregar las actividades metodológicas al sistemas que serán las que se tomaran en cuenta a la hora de realizar los controles.

Observaciones: Solo usuarios con privilegios correspondientes a estas funcionalidades las pueden ejecutar y se debe verificar la integridad de los datos.

Tabla 12: HU 10: Gestionar actividades metodológicas

En la **Tabla 19** se muestra la HU 17: Gestionar Evaluación del Control

Historia de Usuario	
Número: 17	Nombre Historia: Gestionar Evaluación del Control
Usuario: Vicerrector, Controlador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración Asignada: 5
Programador responsable: Wuilliam Hernández Rodríguez	
Descripción: Mostrar, eliminar, editar y agregar evaluación del control al sistema.	
Observaciones: Solo usuarios con privilegios correspondientes a estas funcionalidades las pueden ejecutar y se debe verificar la integridad de los datos.	

Tabla 17: HU 17: Gestionar evaluación del control

2.13 Prototipo de interfaz de usuario

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

Reglamento

Artículo 26

Artículo 40

Artículo 90

Artículo 91

Artículo 93

Artículo 94

Artículo 95

UNIVERSIDAD DE MATANZAS 1972

Control a las Actividades Docentes

Usuario
Escriba su nombre de usuario
wulliam ✓

Contraseña

Entrar

Artículo 26:
El trabajo metodológico se concreta, fundamentalmente, en el desarrollo con calidad del proceso docente educativo, logrando una adecuada integración de las clases con la actividad investigativa y laboral, así como con las tareas de alto impacto social y demás tareas de carácter extracurricular que cumplen los estudiantes. Las funciones principales del trabajo metodológico son la planificación, la organización, la regulación y el control del proceso docente educativo. El adecuado desempeño de estas funciones, que tienen como sustento esencial lo didáctico, garantiza el eficiente desarrollo del proceso docente educativo.

Artículo 40:
El colectivo de año tendrá como principales funciones, la conducción y el control sistemático de la marcha del proceso docente educativo y del cumplimiento de los proyectos educativos de los grupos que conforman el año, desarrollando acciones para eliminar las deficiencias detectadas y proponiendo las medidas que permitan el mejoramiento continuo de la calidad de dicho proceso.

2.14 Tareas a desarrollar

Cada Historia de Usuario se transformará en Tareas de Ingeniería (TI) según lo muestra la siguiente tabla

No	Nombre HU	Tarea de Ingeniería	Iteración
1	Diseño y Creación de la base de datos.	1-Diseño de la base de datos. 2-Creación de la base de datos.	1
2	Diseño de la interfaz de usuario.	3- Diseño de la interfaz principal. 4- Diseño de otras interfaces.	1
3	Autenticarse.	5- Autenticarse.	1
4	Gestionar usuarios.	6- Insertar usuario. 7- Eliminar usuario. 8- Editar usuario. 9- Listar usuario.	2
5	Gestionar Categoría Investigativa.	10-Insertar Categoría Investigativa 11-Eliminar Categoría Investigativa 12-Editar Categoría Investigativa 13-Listar Categoría Investigativa	2

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

6	Gestionar Profesional	Categoría	14-Insertar Categoría Profesional 15-Eliminar Categoría Profesional 16-Editar Categoría Profesional 17-Listar Categoría Profesional	2
7	Gestionar Docente	Categoría	18-Insertar Categoría Docente 19-Eliminar Categoría Docente 20-Editar Categoría Docente 21-Listar Categoría Docente	2
8	Gestionar Especialidad		22- Insertar Especialidad 23- Eliminar Especialidad 24-Editar Especialidad 25-Listar Especialidad	2
9	Gestionar Profesor		26- Insertar Profesor 27- Eliminar Profesor 28-Editar Profesor 29-Listar Profesor	3
10	Gestionar Metodológicas	Actividades	30- Insertar Actividades Metodológicas 31- Eliminar Actividades Metodológicas 32-Editar Actividades Metodológicas 33-Listar Actividades Metodológicas	3
11	Gestionar Facultad		34- Insertar Facultad 35- Eliminar Facultad 36-Editar Facultad 37-Listar Facultad	4
12	Gestionar Departamento		38- Insertar Departamento 39- Eliminar Departamento 40-Editar Departamento 41-Listar Departamento	4
13	Gestionar Carrera		42- Insertar Carrera 43- Eliminar Carrera 44-Editar Carrera 45-Listar Carrera	4
14	Gestionar Grupo		46- Insertar Grupo	4

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

		47- Eliminar Grupo 48-Editar Grupo 49-Listar Grupo	
15	Gestionar Asignatura	50- Insertar Asignatura 51- Eliminar Asignatura 52-Editar Asignatura 53-Listar Asignatura	4
16	Gestionar Tipo de Clase	54- Insertar Tipo de Clase 55- Eliminar Tipo de Clase 56-Editar Tipo de Clase 57-Listar Tipo de Clase	4
17	Gestionar Evaluación del Control	58- Insertar Evaluación del Control 59- Eliminar Evaluación del Control 60-Editar Evaluación del Control 61-Listar Evaluación del Control	5
18	Generar Reportes de Evaluación del Control	62- Generar reporte de Evaluación de Control	6
19	Gestionar Trazas	63-Listar traza 64-Buscar traza	6

Tabla 22: Tareas de Ingeniería

En las siguientes tablas que se muestran las tareas de ingeniería que tenían mayor peso en el desarrollo de esta investigación.

En la **Tabla 23** se muestra la TI 1: Diseño de la base de datos.

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 1
Nombre de Tarea: Diseño de la base de datos.	
Tipo de Tarea: Diseño.	Puntos Asignados: 1
Programador responsable: Wuilliam Hernández Rodríguez	
Descripción: : Analizar con profundidad el negocio referente a la gestión de la información de los proyectos de investigación de la Universidad de Matanzas y luego diseñar la base de datos que permita almacenar dicha información.	

Tabla 23: TI 1: Diseño de la base de datos

En la **Tabla 24** se muestra la TI 2: Creación de la base de datos.

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

Tarea de Ingeniería	
Número de Tarea: 2	Número de Historia de Usuario: 1
Nombre de Tarea: Creación de la base de datos.	
Tipo de Tarea: Desarrollo.	Puntos Asignados: 0.5
Programador responsable: Wuilliam Hernández Rodríguez	
Descripción: Crear en MySQL la base de datos con su respectivas relaciones y la integridad correspondiente entre las tablas.	

Tabla 24: TI 2: Creación de la base de datos

En la **Tabla 50** se muestra la TI 30: Insertar Actividades Metodológicas.

Tarea de Ingeniería	
Número de Tarea: 30	Número de Historia de Usuario: 10
Nombre de Tarea: Insertar Actividades Metodológicas	
Tipo de Tarea: Desarrollo.	Puntos Asignados: 1
Programador responsable: Wuilliam Hernández Rodríguez.	
Descripción: Se mostrará un formulario donde el usuario con los permisos necesarios podrá insertar una actividad demostrativa	

Tabla 50: TI 30: Insertar actividades metodológicas

En la **Tabla 51** se muestra la TI 31: Eliminar Actividades Metodológicas.

Tarea de Ingeniería	
Número de Tarea: 31	Número de Historia de Usuario: 10
Nombre de Tarea: Eliminar Actividades Metodológicas	
Tipo de Tarea: Desarrollo.	Puntos Asignados: 0.5
Programador responsable: Wuilliam Hernández Rodríguez.	
Descripción: Se mostrará una alerta para confirmar la eliminación de la actividad metodológica.	

Tabla 51: TI 31: Eliminar actividades metodológicas

En la **Tabla 52** se muestra la TI 32: Editar Actividades Metodológicas.

Tarea de Ingeniería	
Número de Tarea: 32	Número de Historia de Usuario: 10
Nombre de Tarea: Editar Actividades Metodológicas	
Tipo de Tarea: Desarrollo.	Puntos Asignados: 1
Programador responsable: Wuilliam Hernández Rodríguez.	

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

Descripción: Se mostrará un formulario donde el usuario con los permisos necesarios podrá editar la actividad metodológica.

Tabla 52: TI 32: Editar actividades metodológicas

En la **Tabla 53** se muestra la TI 33: Listar Actividades Metodológicas.

Tarea de Ingeniería	
Número de Tarea: 33	Número de Historia de Usuario: 10
Nombre de Tarea: Listar Actividades Metodológicas	
Tipo de Tarea: Desarrollo.	Puntos Asignados: 0.5
Programador responsable: Wuilliam Hernández Rodríguez.	
Descripción: Se mostrará una tabla <i>que</i> permitirá ver el listado de las actividades metodológicas existentes en la aplicación. Se muestra las opciones de insertar, eliminar y editar. Al pie de tabla se observa el paginador.	

Tabla 53: TI 33: Listar actividades metodológicas.

2.14.1 Tarjetas de Clase, Responsabilidad y Colaboración

La principal funcionalidad que tienen las tarjetas de Clase, Responsabilidad y Colaboración (CRC) es mostrar las colaboraciones que se establecen entre las clases que componen la aplicación. Cada tarjeta representa una clase con su nombre en la parte superior.

A continuación, se muestran algunas de las tarjetas que se elaboraron.

En las tablas se muestran las Tarjetas CRC para las clases: **actividad**, **factor**, **atributo** y **criterio de medida**. En la **Tabla 85** se muestra la Tarjeta CRC para la clase: **actividad**

Tarjeta CRC	
Clase: actividad	
Superclase :	
Subclase:	
Descripción: En esta clase se guardan los datos relacionados solamente con las actividades .	
Atributos	
Idactividad	int
Actividad	text
Maxtotal	int

Tabla 85: tarjeta CRC: actividad

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

En la **Tabla 86** se muestra la Tarjeta CRC para la clase: **factor**

Tarjeta CRC	
Clase: factor	
Superclase :	
Subclase:	
Descripción: En esta clase se guardan los datos relacionados solamente con los factores.	
Atributos	
Idfactor	int
Factor	text
Pesofactor	int

Tabla 86: tarjeta CRC: factor

En la **Tabla 87** se muestra la Tarjeta CRC para la clase: **atributo**

Tarjeta CRC	
Clase: atributo	
Superclase :	
Subclase:	
Descripción: En esta clase se guardan los datos relacionados solamente con los atributos.	
Atributos	
Idatributo	int
Atributo	text
Pesoatributo	int

Tabla 87: tarjeta CRC: atributo

En la **Tabla 88** se muestra la Tarjeta CRC para la clase: **creiteriomedida**

Tarjeta CRC	
Clase: criteriomedida	
Superclase :	
Subclase:	
Descripción: En esta clase se guardan los datos relacionados solamente con los criterios de medidas.	
Atributos	

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

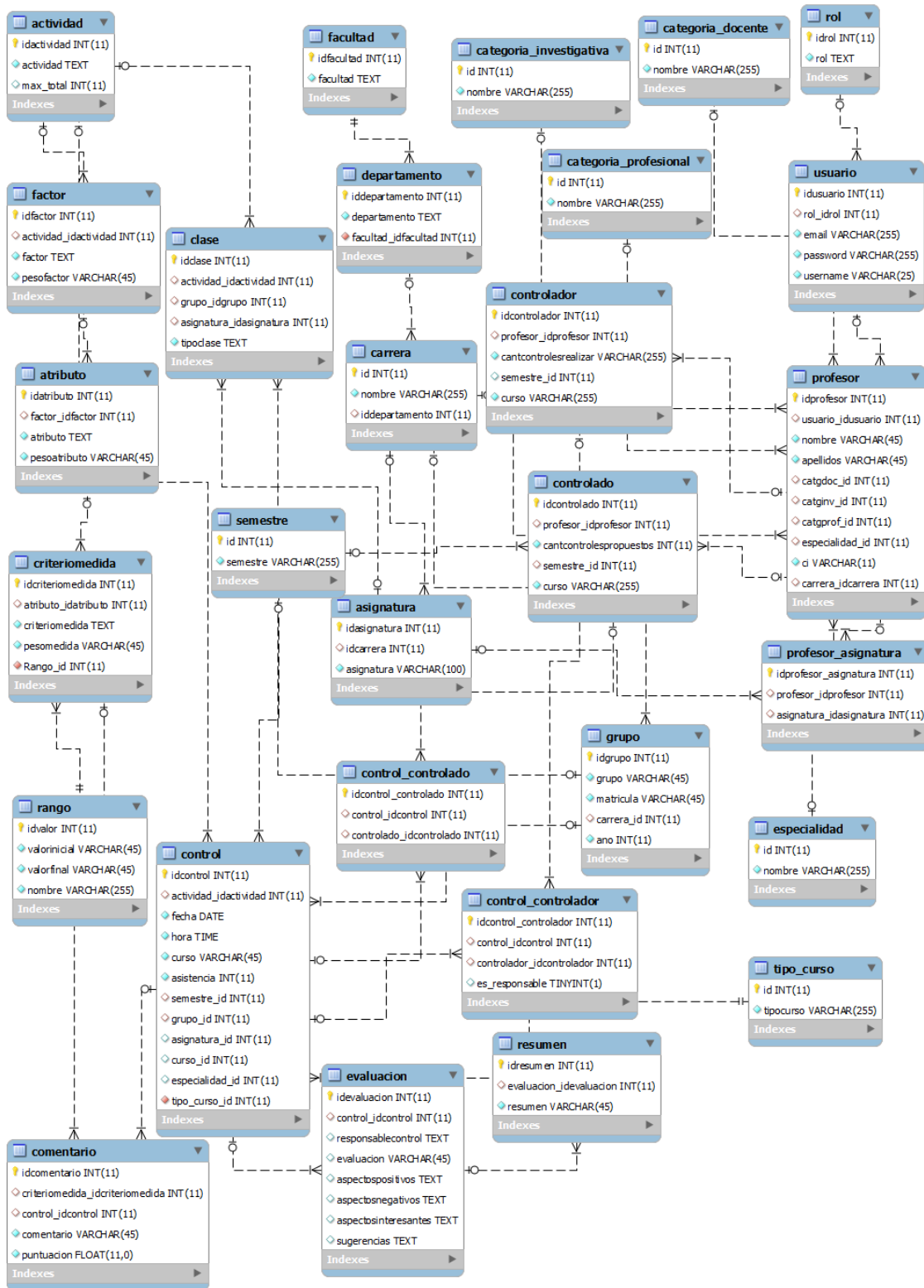
Idcriteriomedia	int
Criteriomedia	text
Pesocriterio	int

Tabla 88: tarjeta CRC: criteriomedia

CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

2.14.2 Modelo físico de la base de datos

En la **Figura 8** se observa el modelo físico de la base de datos de la aplicación web.



CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

2.15 Conclusiones parciales

En este capítulo se plantean las etapas necesarias para desarrollar el software según la metodología XP, con la excepción de las pruebas funcionales. Se definió el equipo de trabajo. Se crearon las historias de usuarios y las tareas de ingeniería planificadas en cada una de las iteraciones. La planificación del proyecto y la estimación de los costos se llevaron a cabo mediante el modelo de Puntos de Función. Los elementos tratados en este capítulo sirvieron para llegar a un acuerdo entre las partes interesadas en el diseño y la estructura de la aplicación a través de la implementación de las funcionalidades que permitan realizar la gestión de la información de los proyectos de investigación de la Universidad de Matanzas.

Capítulo 3: Validación de la solución propuesta

3.1 Introducción

En este capítulo se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el diseño e implementación de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo. Obviamente, la mejor forma de que las pruebas estén correctas es incluir al cliente en el diseño de las mismas.

3.2 Pruebas al software

Para comprobar el cumplimiento de los requisitos funcionales iniciales, se realizan las pruebas al software permitiendo conocer la calidad de este producto, lo que constituye uno de los pasos más importantes en el diseño e implementación de un sistema. Las pruebas persiguen como objetivo, llevar a cabo el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar al menos un error no descubierto hasta entonces. Los niveles de trabajo en los cuales se pueden realizar las pruebas son:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Sistema.
- Prueba de Aceptación.
- Prueba de Seguridad.

3.2.1 Plan de pruebas

Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente.

A continuación, se muestra en la **Tabla 106** el plan de pruebas.

No.	Historia de Usuario	Pruebas a realizar
-----	---------------------	--------------------

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

1	Diseño y Creación de la base de datos.	1-Test base de datos.
2	Diseño de la interfaz de usuario.	2-Test diseño de la interfaz de usuario.
3	Autenticarse.	3-Test autenticarse.
4	Gestionar usuario.	4-Test Insertar usuario. 5-Test Eliminar usuario. 6-Test Editar usuario. 7-Test Listar usuario.
5	Gestionar Categoría Investigativa	8-Test Insertar Categoría Investigativa. 9-Test Eliminar Categoría Investigativa. 10-Test Editar Categoría Investigativa. 11-Test Listar Categoría Investigativa.
6	Gestionar Categoría Profesional	12-Test Insertar Categoría Profesional. 13-Test Eliminar Categoría Profesional. 14-Test Editar Categoría Profesional. 15-Test Listar Categoría Profesional.
7	Gestionar Categoría Docente	16-Test Insertar Categoría Docente. 17-Test Eliminar Categoría Docente. 18-Test Editar Categoría Docente. 19-Test Listar Categoría Docente.
8	Gestionar Especialidad	20-Test Insertar Especialidad. 21-Test Eliminar Especialidad. 22-Test Editar Especialidad. 23-Test Listar Especialidad.
9	Gestionar Profesor	24-Test Insertar Profesor. 25-Test Eliminar Profesor. 26-Test Editar Profesor. 27-Test Listar Profesor.
10	Gestionar Actividades Metodológicas	28-Test Insertar Actividades Metodológicas. 29-Test Eliminar Actividades Metodológicas. 30-Test Editar Actividades Metodológicas. 31-Test Listar Actividades Metodológicas.
11	Gestionar Facultad	32-Test Insertar Facultad.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

		33-Test Eliminar Facultad. 34-Test Editar Facultad. 35-Test Listar Facultad.
12	Gestionar Departamento	36-Test Insertar Departamento. 37-Test Eliminar Departamento. 38-Test Editar Departamento. 39-Test Listar Departamento.
13	Gestionar Carrera	40-Test Insertar Carrera. 41-Test Eliminar Carrera. 42-Test Editar Carrera. 43-Test Listar Carrera.
14	Gestionar Grupo	44-Test Insertar Grupo. 45-Test Eliminar Grupo. 46-Test Editar Grupo. 47-Test Listar Grupo.
15	Gestionar Asignatura	48-Test Insertar Asignatura. 49-Test Eliminar Asignatura. 50-Test Editar Asignatura. 51-Test Listar Asignatura.
16	Gestionar Tipo de Clase	52-Test Insertar Tipo de Clase. 53-Test Eliminar Tipo de Clase. 54-Test Editar Tipo de Clase. 55-Test Listar Tipo de Clase.
17	Gestionar Evaluación del Control	56-Test Insertar Evaluación del Control. 57-Test Eliminar Evaluación del Control. 58-Test Editar Evaluación del Control. 59-Test Listar Evaluación del Control.

Tabla 106: Plan de Pruebas

3.2.3 Pruebas Funcionales:

Insertar Actividades Metodológicas

Atributos:

- actividad
- factores

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

- pesofactor
- atributos
- pesoatributo
- criteriomedida
- pesomedida

1-Clases de equivalencia por cada atributo:

Atributo	Válida	Representante	Inválida	Representante
actividad	1-Cualquier combinación de letras que comience con mayúscula.	Control a Clases	2-Que este vacía	NULL
			3-Cadena que no comience con mayúscula.	Control a clases
factor	4- Cualquier combinación de letras.	Trabajo Político Ideológico	5-Que este vacía	NULL
Pesofactor	6- Un numero entero entre 1 y5	3	7- Que este vacía	NULL
			8- Que el numero sea mayor que 5	9
			9- Que el numero sea menor que 1	0
atributo	10- Cualquier combinación de letras.	Educación a través de la instrucción	11-Que este vacía	NULL
Pesoatributo	12- Un numero entero entre 1 y5	1	13- Que este vacía	NULL
			14- Que el numero sea mayor que 5	6
			15- Que el numero sea menor que 1	0
criteriomedida	16- Cualquier combinación de letras.	Puntualidad del profesor	17-Que este vacía	NULL
Pesomedida	18- Un numero entero entre 1 y5	1	19- Que este vacía	NULL
			20- Que el numero sea mayor que 5	7

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

21- Que el numero sea menor que 1	0
-----------------------------------	---

2-Definir casos de Pruebas:

No.	Clase Equiv.	actividad	factor	Peso factor	atributo	Peso atributo	criterio medida	Peso medida	Resultado Esperado
1	1,4,6,10,12,16,18	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	1	Puntualidad del profesor	1	Resultado esperado
2	2,4,6,10,12,16,18	NULL	Trabajo Político Ideológico	3	Educación a través de la instrucción	1	Puntualidad del profesor	1	La actividad no puede estar vacía
3	1,5,6,10,12,16,18	Control a Clases	NULL	3	Educación a través de la instrucción	1	Puntualidad del profesor	1	El factor no puede estar vacío
4	1,4,7,10,12,16,18	Control a Clases	Trabajo Político Ideológico	NULL	Educación a través de la instrucción	1	Puntualidad del profesor	1	El peso del factor no puede estar vacío
5	1,4,8,10,12,16,18	Control a Clases	Trabajo Político Ideológico	9	Educación a través de la instrucción	1	Puntualidad del profesor	1	El peso del factor no puede ser mayor que 5
6	1,4,9,10,12,16,18	Control a Clases	Trabajo Político Ideológico	0	Educación a través de la instrucción	1	Puntualidad del profesor	1	El peso del factor no puede estar vacío
7	1,4,6,11,12,16,18	Control a Clases	Trabajo Político Ideológico	3	NULL	1	Puntualidad del profesor	1	El atributo no puede estar vacío
8	1,4,6,10,13,16,18	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	NULL	Puntualidad del profesor	1	El peso del atributo no puede estar vacío

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

9	1,4,6,10,14,16,18	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	9	Puntualidad del profesor	1	El peso del atributo no puede ser mayor que 5
10	1,4,6,10,15,16,18	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	0	Puntualidad del profesor	1	El peso del atributo no puede estar vacío
11	1,4,6,10,12,16,18	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	1	NULL	1	El criterio no puede estar vacío
12	1,4,6,10,12,17,19	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	1	Puntualidad del profesor	NULL	El peso del criterio no puede estar vacío
13	1,4,6,10,12,16,20	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	1	Puntualidad del profesor	9	El peso del criterio no puede ser mayor que 5
14	1,4,6,10,12,16,21	Control a Clases	Trabajo Político Ideológico	3	Educación a través de la instrucción	1	Puntualidad del profesor	0	El peso del criterio no puede estar vacío

3- Tabla de Prueba por Caso de Prueba:

No.	1
Requerimiento	Adicionar actividad
Objetivo	Probar la acción de adicionar actividad(clases equivalentes 1)
Tipo de Prueba	Funcional
Hardware	Un procesador A10 , disco duro de 240GB SSD, memoria RAM de 12GB
Software	Sistema Operativo Windows 10, Base de Datos Xampp, Navegador Mozilla Firefox 59

Personal	Ingeniero de Pruebas		
Casos de Prueba			
Datos de Entrada	Nombre: Control a clases		
Resultados Esperados	Mensaje: "Operación exitosa "		
Resultados Obtenidos	Si() No()		
Casos de Excepción:		Comentarios:	
Aprobado por: Wuilliam Hernández Rodríguez	Cargo: pruebas	Líder de	Fecha:

Tabla 28: Caso de Prueba #1

3.3 Herramientas de Pruebas.

También se emplearon herramientas para hacer pruebas tal es el caso del software Subgraph Vega y Webstress obteniendo los siguientes resultados:

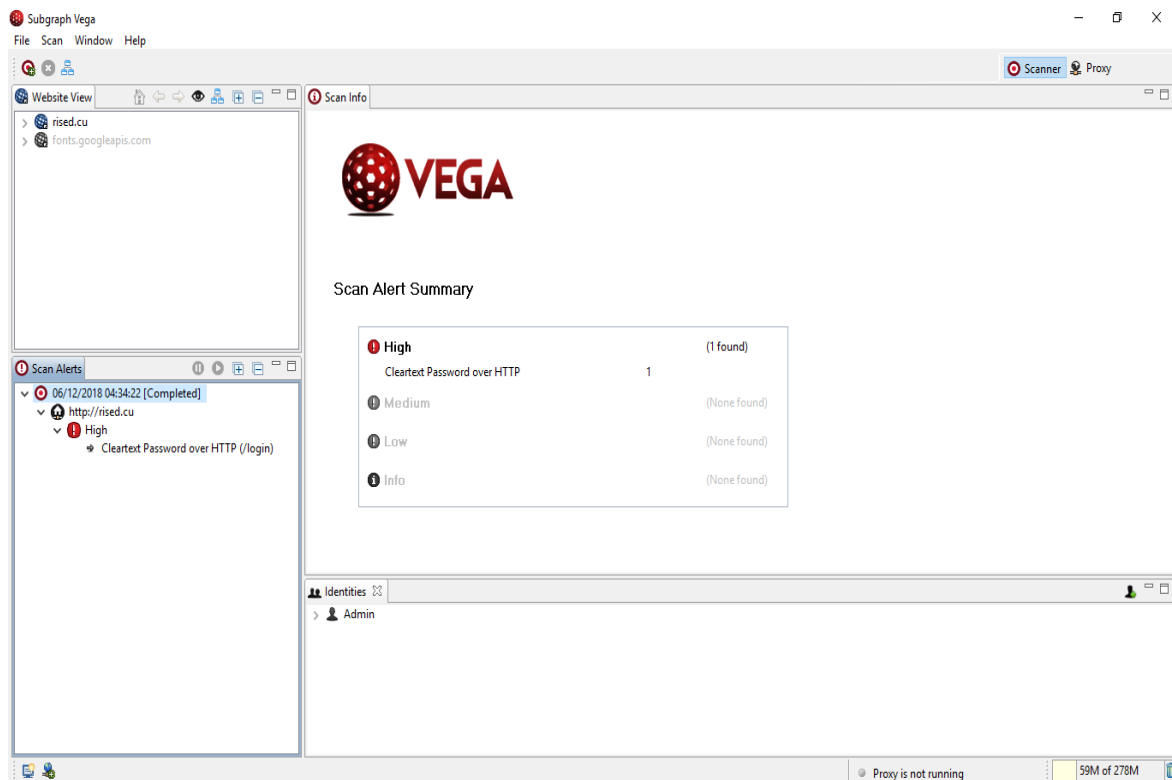


Figura 5: Resultado de una prueba realizada con Subgraph Vega.

3.4 Análisis de los resultados obtenidos.

Al concluir el desarrollo del proceso de pruebas se lograron resultados satisfactorios en el progreso del sistema con un nivel medio de sencillez. Las

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

pruebas fueron desplegadas por cada una de las historias de usuario. Los elementos de pruebas abordados, permitieron validar el funcionamiento de la aplicación y los resultados satisfactorios de dichas pruebas.

3.5 Conclusiones Parciales.

Las pruebas realizadas utilizando las técnicas anteriormente planteadas fueron de gran importancia para demostrar el buen funcionamiento del software y el cumplimiento de los requisitos planteados por el cliente. Una vez realizadas las pruebas se logró brindarle al cliente una versión del software completamente funcional que facilitara la informatización de la gestión de los riesgos en los softwares educativos.

Conclusiones Generales

En esta investigación quedaron satisfechos los objetivos planteados y de esta forma se determinaron las siguientes conclusiones:

- Se determinaron el marco teórico referencial sobre la gestión de la información de los riesgos durante el proceso de desarrollo de los softwares educativos que se exponen a continuación:
 - Se normalizaron los fundamentos teóricos que permitan la creación de un sistema web que gestione la información asociada a los controles a las actividades docentes.
 - Se determinó la metodología eXtreme Programming (XP) para el desarrollo del sistema web debido a sus características.
 - Se determinaron como herramientas más viables para la creación del proyecto el gestor de base de datos MySQL 4.5.1, lenguaje PHP 5.6.21, como servidor web Apache 2.4.17, *framework* Symfony 2.7.3, el Entorno de Desarrollo Integrado (IDE) utilizado fue JetBrains PhpStorm 2016.1
- Se implementó una aplicación web que facilite la gestión de los controles a las actividades docentes.
- Se aplicó un sistema de pruebas que permitió detectar errores cometidos durante los procesos anteriores del desarrollo del sistema web para la gestión de los controles a las actividades docentes y corregirlos.

De forma general, se concluye que la aplicación web desarrollada es una herramienta confiable y fácil de utilizar pues eleva la calidad de la gestión de los controles a las actividades docentes, reduce los errores a cometer, facilita y perfecciona el proceso y brinda reportes de interés a los directivos. Por estas razones, queda respondida la pregunta científica planteada en la introducción.

RECOMENDACIONES

Recomendaciones

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

1. Aprovechar las posibilidades de información almacenadas en la aplicación e implementar funcionalidades como la notificación de nuevos controles realizados y la gestión de controles asignados a los profesores controlados y controladores.
2. Agregarle nuevos reportes que sean de interés para los usuarios que interactúen con la aplicación.
3. También se recomienda poner en funcionamiento el sistema web para automatizar la gestión del control a las actividades docentes.

BIBLIOGRAFÍA

Bibliografía

- Achour, M. (2007). *Manual de PHP*.
- Alcalde, A. (2013). *El baúl del programador*. Obtenido de <http://elbauldelprogramador.com/los-10-mejores-frameworks-gratis-de-aplicaciones-web/>
- Álvarez, M. Á. (30 de 8 de 2007). *DesarrolloWeb*. Obtenido de <http://desarrolloweb.com>
- Álvarez, R. (2012). *Introducción al HTML*. Obtenido de <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
- Arocha, A. T. (2016). *Aplicación web para la gestión de equipos en depósito en la EMAE Matanzas*. Matanzas.
- Betancourt Perera, R. (2016). *Sistema web para generar mapas conceptuales*.
- Bowler, T., & Bancner, W. (2009). *Symfony 1.3 web application development*.
- Chaffer, J. (2009). *Learning JQuery 1.3: Better Interaction and Web Development with 21 Simple JavaScript Techniques*.
- Cristian. (2008). *Ventajas de usar CSS*. Obtenido de <http://www.stardustxs.com/2008/03/05/ventajas-de-usar-css/>
- Eguiluz, J. (2014). *Desarrollo de Web Agil con Symfony2*.
- Falcón, C. (2006). *An Epic Review of PyCharm 3. Vim User's Perspective*.
- Fernández, P. (2011). *JetBrains anuncia PHP Storm*.
- Fernández, R. A. (29 de 04 de 2016). *Procedimiento para la atención de los equipos de la reserva estatal*.
- Fuentes, J. (2015). *Desarrollo de Software Ágil. Extreme Programming y Scrum*.
- García, D. C. (2016). *Aplicación informática para gestión de reservas y estancias Hostal Azul Mtzas*. Matanzas.
- Garrett, J. (2005). *Ajax: A new approach to web applications*.
- Gauchat, J. D. (2012). *El Gran libro de HTML5, CSS3 y Javascript*. España.
- Joskowicz, J. (2008). *Reglas y Prácticas en Extreme Programming*.
- Loaiza, V., & Zorro, L. (2010). *Plan de Pruebas de Software*.
- Luque, E. G. (2012). *Informatización del control de las reparaciones y mantenimientos de los vehículos automotores en la UEB Municipal de la Empresa Eléctrica de Matanzas*. Matanzas.
- Marcano, H. (2009). *Desarrollo de una aplicación educativa bajo ambiente web, como apoyo para la enseñanza de la asignatura bases de datos orientados a objetos, para la carrera de Licenciatura en Informática de la Universidad de Oriente*.
- Martínez, D. (2014). *PostgreSQL vs MySQL*. Obtenido de [Http://danielpecos.com/documents/postgresql-vs-mysql/](http://danielpecos.com/documents/postgresql-vs-mysql/)

BIBLIOGRAFÍA

- Meléndez, K., Abraham, D., & Carla, B. (30 de 01 de 2005). Estimación del Esfuerzo en la etapa Post-Arquitectura usando el modelo COCOMO II Versión 2.0.
- Mondéjar, A. G. (2015). *Aplicación Web para el apoyo a los ajustes de planes de estudio en la Universidad de Matanzas*. Matanzas.
- Peña, M. Á. (2007). *Arquitectura cliente-servidor*. Obtenido de <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
- Pérez, F. F. (2015). *Sistema Informático para control de trabajo en taller de chapistería y pintura CUBACAR*. Matanzas.
- Pérez, J. (2009). *Introducción a JavaScript*.
- Pressman. (2010). *Ingeniería del software. Un enfoque práctico*. Mexico.
- Quintero, E. (2012). *Sistema de gestión de la información para la secretaria general de la Universidad de Matanzas "Camilo Cienfuegos"*. Matanzas.
- Resolucion, 20 (11 de 04 de 2013).
- Resolución 4 (SIME 2003).
- Revuelta, A. B. (2015). *Aplicación Informática para la Gestión de Equipos y Piezas de Repuesto de Importación de la CTE Antonio Guiteras*. Matanzas.
- Risueño, P. (2013). *Comenzando con Bootstrap, framework responsive*.
- Rivero, O. (2015). *Sistema Informático para la automatización del análisis de indicadores económicos por medio de redes neuronales*. Matanzas.
- RocketTheme. (2009). *Metodologías de desarrollo de software*.
- Rodríguez, J. A. (2014). *Sistema Informático para la gestión de estudiantes internos y control de capacidades en la Residencia Estudiantil de la*. Matanzas.
- Sánchez. (2003). *Diferencias entre metodologías ágiles y no ágiles*.
- Sánchez, T. P., & Letelier. (2003). *Metodologías Ágiles en el desarrollo de Software*.
- Schwartz, B. (2012). *High Performance MySQL*.
- Seguridad para todos. (2012). *Seguridad para todos*. Obtenido de <http://www.seguridadparatodos.es/2012/02/vega-nueva-solucion-open-source-para.html>
- Senso, J. A. (2015). *Servidores WAMP para producción local*. Obtenido de <http://tecnologiasweb.jsenso.es/servidores-wamp/>
- Tamayo, S. A. (2016). *Mantenimiento perfecto correctivo a la aplicación web Ajustes-UM*. Matanzas.
- Tedeschi, N. (2013). *¿Qué es un Patrón de Diseño?*
- Trey-SAT. (15 de 11 de 2016). *Trey-SAT*. Obtenido de <http://http://www.portalprogramas.com/trey-sat/>
- V&V Quality. (2016). *V&V Quality*. Obtenido de <http://vyvquality.com/pruebas-seguridad/>

BIBLIOGRAFÍA

Zanotti, A. (2016). *El software libre su difusión en Argentina: mercado, estado, sociedad.*

Anexos

Anexo 1: Tablas de ponderaciones para EI, EQ, EO, ILF, EIF

CLASIFICACIÓN DE ENTRADAS Y CONSULTAS	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

CLASIFICACIÓN DE SALIDAS	1-5 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 o 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 3 ficheros accedidos	MEDIA 5	ALTA 7	ALTA 7

FICHEROS LÓGICOS INTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 5 Entidades o registros lógicos	MEDIA 10	ALTA 15	ALTA 15

FICHEROS LÓGICOS EXTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 5	BAJA 5	MEDIA 7
2 - 5 Entidades o registros lógicos	BAJA 5	MEDIA 7	ALTA 10
Más de 5 Entidades o registros lógicos	MEDIA 7	ALTA 10	ALTA 10

ANEXOS

Anexo 2: Tabla para obtener el ACT

No.de Factor	Factor de Ajuste	Min	Max
1	Comunicación de Datos	0	5
2	Proceso Distribuido	0	5
3	Objetivos de Rendimiento	0	5
4	Configuración de Explotación Compartida	0	4
5	Tasa de transacciones	0	5
6	Entrada de Datos en Línea	0	5
7	Eficiencia con el Usuario Final	0	5
8	Actualizaciones en Línea	0	5
9	Lógica de Proceso Interno Compleja	0	5
10	Reusabilidad del Código	0	5
11	Conversión e Instalación contempladas	0	5
12	Facilidad de Operación	0	5
13	Instalaciones Múltiples	0	5
14	Facilidad de Cambios	0	5

Anexo 3: Tabla para obtener las Líneas por Puntos de Función

Entorno y Lenguaje	Líneas de Código por PF	Horas por PF
Lenguajes 2GL: Ensamblador, C,...	300	20 a 30
Lenguajes 3GL: Cobol	100	10 a 20
Lenguajes 4GL: VisualXX	20	5 a 10

Anexo 4: Figura # Mapa de procesos de control a las actividades docentes.
Fuente: Elaboración del autor

Diagrama de proceso del negocio (General)

