

**Universidad de Matanzas**



**Facultad de Ciencias Técnicas  
Especialidad Ingeniería Informática**

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

**Aplicación web en apoyo a la gestión de la información del Colectivo de  
Carrera de la Universidad de Matanzas**

**Autor: Lorena Varela García**

**Tutor: Walfredo González Hernández**

**Matanzas, 2018**

*"Al mundo nuevo corresponde la Universidad  
nueva".*

*José Martí*

# *Dedicatoria*

A mi mamá que le debo todo lo que soy.

## *Agradecimientos:*

- ❖ A mi mamá, la amo con todo mi corazón y sin su fuerza no hubiera llegado tan lejos.
- ❖ A Etien por su apoyo incondicional que lo quiero como un padre.
- ❖ A mi hermana querida Laura por estar siempre y darme ánimo en las dificultades.
- ❖ A mi abuela Andrea que desde lejos siempre estuvo pendiente de mis estudios y orgullosa de su nieta.
- ❖ A toda mi familia por haber creído en mí y darme esperanzas.
- ❖ A mis compañeras de aulas Lianet, Zuliam y Heydi, por su amistad y ayudarme cuando las necesitaba.
- ❖ A todos mis compañeros de aula por su ayuda sincera y comprensión en todo momento, por el tiempo que compartimos juntos, los voy a extrañar.
- ❖ A mi tutor: Walfredo Gonzáles Hernández por sus enseñanzas, por sus consejos.
- ❖ A los profesores de la Universidad y del Departamento de Informática que durante esta etapa me formaron como profesional.

A todos,  
Muchas Gracias

**Opinión del Tutor del Trabajo de Diploma.**

Declaración de autoría

Yo, Lorena Varela García, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas “Camilo Cienfuegos”, especialmente a la Facultad de Ciencias Técnicas a que hagan el uso que estimen pertinente de él.

Y para que así conste, firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del 2018.

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Tutor

## **Resumen**

En el presente trabajo se describe la concepción e implementación de una aplicación web para la gestión de la información del Colectivo de Carrera en la Universidad de Matanzas con el fin de elevar la calidad del proceso docente educativo y garantizar la organización de los diferentes niveles de dirección. El sistema automatizado tiene el propósito de proveer a la entidad de una herramienta eficiente que permita registrar, procesar y controlar de forma integrada toda la información correspondiente, lo que propiciará un mayor control, ahorro de tiempo y recursos.

La investigación realizada posee un aporte práctico en la recopilación, almacenamiento, procesamiento y gestión, de forma eficiente y segura de todas las informaciones referentes al Colectivo de Carrera en la Universidad de Matanzas; lo que permite garantizar una eficiente planificación del flujo de trabajo y realizar análisis profundos y necesarios para la toma de decisiones de forma tal que se eviten errores que se comenten en la persistencia, manipulación y extracción de la información.

Para lograr una solución responsable se emplearon técnicas de Ingeniería de Software, de Programación orientada a la Web y de almacenamiento de información basada en motores de bases de datos que demostraron ser competentes para el logro del objetivo de la investigación.

## **Abstract**

This paper describes the conception and implementation of a web application for the management of the information of the Career Collective at the University of Matanzas in order to raise the quality of the educational process and ensure the organization of the different levels of management and organizational. The automated system has the purpose of providing the entity with an efficient tool that allows to register, process and control in an integrated manner all the corresponding information, which will lead to greater control, saving time and resources.

The research carried out has a practical contribution in the collection, storage, processing and management, in an efficient and safe way of all the information related to the Career Collective in the University of Matanzas; what allows to guarantee an efficient planning of the flow of work and to carry out deep and necessary analyzes for the decision making in such a way that errors are avoided that are commented on the persistence, manipulation and extraction of the information.

To achieve a responsible solution, techniques of Software Engineering, Web-oriented Programming and storage of information based on database engines were used, which proved to be competent to achieve the objective of the research



## Índice general

<b>Introducción</b> .....	13
<b>Capítulo 1. Marco Teórico Referencial</b> .....	16
1.1 Introducción.....	16
1.2 Objeto de estudio.....	16
1.2.1 Flujo actual de los procesos de información .....	17
1.2.2 Descripción de los procesos que serán automatizados .....	17
1.3 Antecedentes de la investigación .....	17
1.4 Metodología de la investigación.....	18
1.4.1 Métodos Teóricos Utilizados .....	18
1.4.2 Métodos Empíricos Utilizados .....	18
1.5 Fundamentación de la metodología utilizada.....	18
1.5.1 Metodología de desarrollo de software: XP .....	19
1.6 Herramientas y tecnologías utilizadas .....	20
1.6.1 Tendencias tecnológicas a considerar .....	20
1.6.2 Lenguajes de programación:.....	22
1.6.3 Framework de desarrollo web .....	24
1.7 Conclusiones del capítulo.....	25
<b>Capítulo 2: Descripción de la solución propuesta.</b> .....	26
2.1 Introducción al capítulo .....	26
2.2 Descripción de la solución .....	26
2.3 Etapa de planificación.....	26
2.3.1 Equipo de trabajo y roles.....	27
2.3.2 Historias de Usuario Iniciales .....	27
2.3.3 Plan de Iteraciones.....	29
2.3.4 Análisis del costo.....	30

2.3.5 Análisis de los beneficios .....	32
2.3.6 Plan de Entregas.....	32
2.4 Etapa de diseño.....	33
2.4.1 Prototipo de la interfaz de usuario.....	34
2.4.2 Tareas de Ingeniería .....	39
2.4.3 Modelo físico de la Base de datos.....	43
2.4.4 Tarjetas de Clase, Responsabilidad y Colaboración .....	43
2.5 Conclusiones parciales del Capítulo.....	44
<b>Capítulo 3: Validación de la solución propuesta.....</b>	<b>45</b>
3.1 Introducción .....	45
3.2 Pruebas al software .....	45
3.2.1 Plan de pruebas .....	45
3.2.2 Pruebas de aceptación.....	47
3.2.3 Pruebas de seguridad .....	54
3.3 Análisis de los resultados obtenidos .....	55
3.4 Conclusiones parciales .....	55
<b>Conclusiones Generales .....</b>	<b>56</b>
<b>Recomendaciones.....</b>	<b>57</b>
<b>Bibliografía.....</b>	<b>58</b>
<b>Anexos .....</b>	<b>60</b>

## Índice de Tablas

Tabla 1 Equipo de trabajo y roles.....	27
Tabla 2 Planificación de las Historias de Usuario.....	29
Tabla 3 Plan de entrega.....	33
Tabla 4 Diseño y creación de la Base de datos .....	35
Tabla 5 Autenticación del Usuario .....	35
Tabla 6 Gestionar Usuarios.....	36
Tabla 7 Gestionar Facultad .....	36
Tabla 8 Gestionar Carrera.....	36
Tabla 9 Gestionar Departamento .....	37
Tabla 10 Gestionar Disciplina.....	37
Tabla 11 Gestionar Disciplina.....	37
Tabla 12 Gestionar Profesor .....	38
Tabla 13 Gestionar Cohorte .....	38
Tabla 14 Gestionar Plan de Proceso Docente PPD .....	38
Tabla 15 Generar Plan de Proceso Docente PPD .....	39
Tabla 16 Generar Solicitud de Servicios .....	39
Tabla 17 Generar Balance de Carga.....	39
Tabla 18 Resumen de las tareas de ingeniería. ....	41
Tabla 19 TI: Diseño de la base de datos.....	41
Tabla 20 TI: Creación de la base de datos.....	41
Tabla 21 TI: Insertar usuario .....	42
Tabla 22 TI: Modificar usuario .....	42
Tabla 23 TI: Eliminar usuario.....	42
Tabla 24 TI: Insertar Plan de Proceso Docente.....	42
Tabla 25 TI: Modificar Plan de Proceso Docente .....	43
Tabla 26 TI: Deshabilitar Plan de Proceso Docente.....	43
Tabla 27 Tarjeta CRC Usuario .....	44
Tabla 28 Plan de Pruebas .....	47

Tabla 29 PA Test Base de datos.....	47
Tabla 30: PA Test Autenticación del usuario.....	48
Tabla 31: PA Test Gestionar asignatura .....	48
Tabla 32: Clases de equivalencia Insertar asignatura.....	49
Tabla 33: Resumen de Casos de prueba .....	51
Tabla 34Test Insertar asignatura. Caso de preba1 .....	51
Tabla 35Test Insertar asignatura. Caso de preba4 .....	52
Tabla 36Insertar asignatura. Caso de prueba13 .....	53

### **Índice de ilustraciones**

Ilustración 1 Plan de Iteraciones .....	30
Ilustración 2 Plan de entregas.....	33
Ilustración 3 Insertar carrera.....	34
Ilustración 4 Insertar Asignatura.....	34
Ilustración 5 Crear Plan de Proceso Docente.....	35
Ilustración 6 Modelo de la base de datos .....	43
Ilustración 7 Test Insertar asignatura. Caso de preba1 .....	51
Ilustración 8Test Insertar asignatura. Caso de preba4.....	52
Ilustración 9Insertar asignatura. Caso de prueba13.....	53
Ilustración 10 Prueba de seguridad. VEGA.....	54

## Introducción

La Informatización de la Sociedad ha sido una de las estrategias prioritarias del Estado cubano desde hace algunos años. El país ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible. El impacto social de las Tecnologías de la Información y las Comunicaciones (TIC) toca muy de cerca a las universidades, propiciando modificaciones en las formas tradicionales de enseñar y aprender.

Uno de los rasgos esenciales que rige el trabajo metodológico en los centros de educación superior lo constituyen los diferentes niveles organizativos del proceso docente; uno de estos niveles, y de importancia ineludible, es el colectivo de carrera. En la educación superior cubana y particularmente en la estructura de las universidades, este constituye el colectivo metodológico con el propósito de contribuir al cumplimiento con calidad del modelo profesional. El trabajo metodológico que se realiza en la carrera se debe caracterizar por su sistematicidad, y debe estar dirigido a coordinar el trabajo de los dos subsistemas fundamentales de la misma: las disciplinas y los años académicos (Milán Licea, y otros, 2017).

Como parte del Proyecto Gestión Universitaria que se desarrolla en la Universidad de Matanzas se estableció como una de las líneas de acción la gestión de la docencia. Su selección estuvo basada en el análisis del funcionamiento de las carreras y de los resultados de la auto-evaluación institucional que reflejaron insuficiencias en la gestión de la carrera. En el **colectivo de carrera de Licenciatura en Educación Informática la Universidad de Matanzas**, a pesar de constituir el eslabón de base de la carrera, no se gestiona ni se almacena información para la toma de decisiones capaz de brindar la organización necesaria. Esto es debido a que la información es manipulada a través de documentos Word y Excel por los coordinadores de carreras y demás miembros, trayendo consigo errores recurrentes, poca planificación y posible pérdida de información.

Es necesario destacar que los problemas que se detectan no pueden resolverse de forma efectiva si no se acude al uso de las nuevas tecnologías para ello es necesario la implementación de un sistema confiable capaz de brindar organización y seguridad al colectivo y a la universidad en general.

Por todo lo anterior se constata la existencia de un problema real a resolver que se convierte en el **problema científico** de la siguiente manera: ¿Cómo facilitar la gestión de la información del Colectivo de Carrera en la Universidad de Matanzas?

Como **objeto de estudio** de esta investigación se tiene la gestión de la información del Colectivo de Carrera en la Universidad de Matanzas

El **campo de acción** es la informatización del proceso de gestión de la información del Colectivo de Carrera en la Universidad de Matanzas

Del problema anterior se deduce la siguiente **hipótesis**: Si se implementa un sistema web entonces se incrementará la gestión de la información del Colectivo de Carrera en la Universidad de Matanzas.

**Objetivo General**: desarrollar una aplicación Web para la gestión de información del Colectivo de Carrera en la Universidad de Matanzas

Los **objetivos específicos** de este trabajo son:

1. Determinar los referentes teóricos – metodológicos sobre la gestión de información del Colectivo de Carrera de la Universidad de Matanzas
2. Determinar las tendencias tecnológicas para la realización de la Aplicación Web para la gestión de información del Colectivo de Carrera de la Universidad de Matanzas
3. Implementar una propuesta de Aplicación Web para la gestión de la Información del Colectivo de Carrera de la Universidad de Matanzas
4. Validar la Aplicación Web para la gestión de información del Colectivo de Carrera de la Universidad de Matanzas

### **Métodos empíricos**

- ✓ Entrevista
- ✓ Estudio de documento
- ✓ Observación

### **Métodos teóricos:**

- ✓ Histórico - Lógico
- ✓ Análisis - Síntesis
- ✓ Inductivo - Deductivo

La tesis está estructurada en una introducción, tres capítulos, las conclusiones, las recomendaciones, la bibliografía y anexos

En el Capítulo 1 se encuentra descrito el **Marco Teórico-Referencial**, se hace una descripción de cada una de las herramientas que se utilizan para la confección del sistema y se explican los métodos científicos empleados.

En el capítulo 2 se realiza el **Análisis, Diseño de la Solución Propuesta y Construcción** donde se argumenta la solución que se propone al problema de investigación, presentando una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software Programación Extrema. Se construye la solución propuesta, presentando una planificación por iteraciones.

En el capítulo 3 se presenta la **Validación de la Solución Propuesta** se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio del clientes y los propios de la metodología de software. Se realiza además un estudio de los beneficios tangibles e intangibles como resultado de la realización del proyecto de software. Las pruebas son una actividad en la cual un sistema es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema. Se realiza la validación a la solución propuesta, con el objetivo de comprobar la eficiencia de las clases u operaciones utilizadas para dar respuesta a los distintos requisitos planteados por el cliente.

Finalmente, se presentan las **Conclusiones y Recomendaciones** de la investigación para dejar el camino abierto a futuros estudios relacionados con la temática abordada.

Asimismo, quedan recogidas las **Bibliografías** empleadas y **Anexos** que fueron necesarios para el desarrollo de todo el trabajo y un mejor entendimiento del mismo.

## Capítulo 1. Marco Teórico Referencial

### 1.1 Introducción

Hoy nuestra universidad tiene el desafío de lograr la calidad y la pertinencia de las carreras que desarrolla mediante la creatividad y flexibilidad curriculares y, por tanto, lograr una universidad responsable. Este capítulo contiene las principales definiciones asociadas al dominio del problema a resolver, los antecedentes del trabajo, el objeto de estudio, los métodos de investigación empleados. Se abordan también los aspectos relacionados con las tendencias tecnológicas, tales como las tecnologías que influyeron en el desarrollo de la aplicación Web y la metodología de ingeniería del software a utilizar, el servidor de base de datos, y los lenguajes utilizados.

### 1.2 Objeto de estudio

El Colectivo de Carrera es el encargado de llevar a cabo el trabajo metodológico en este nivel organizativo. Agrupa a los profesores que dirigen los colectivos de disciplina y de año que integran la carrera en la sede central, a los coordinadores de carrera de las sedes universitarias y a la representación estudiantil. Tiene como propósito lograr el cumplimiento con calidad del modelo del profesional, dirigiendo así el trabajo de las disciplinas y los años. La conducción de este colectivo metodológico está a cargo del jefe de colectivo de carrera (Vela Valdés, y otros, 2007).

Se considera a la carrera como un sistema, integrado por dos subsistemas fundamentales: primero, estructurando verticalmente la carrera, están las disciplinas, y estas como la integración de las asignaturas, las cuales a su vez constituyen integración de temas y estos de clases. De este modo, las asignaturas se imparten en un determinado año, y por su lógica interna forman parte de una determinada disciplina (Milán Licea, et al., 2017).

El colectivo de carrera tendrá como principales funciones:

- Asegurar el cumplimiento de los objetivos del plan de estudio y, en particular, de los objetivos generales de la carrera, proponiendo al decano las acciones necesarias para lograr el mejoramiento continuo de la calidad del proceso docente educativo de la carrera.
- Garantizar el adecuado diseño y aplicación de la estrategia educativa de la carrera, tomando como punto de partida el modelo del profesional.
- Lograr un adecuado balance de las tareas curriculares y extracurriculares que cumplen los estudiantes como parte de su formación integral.



- Participar en el diseño del plan de estudio y adecuarlo a las particularidades del centro y del territorio.
- Evaluar el desempeño del colectivo en el desarrollo del trabajo metodológico de la carrera (Vela Valdés, y otros, 2007).

#### 1.2.1 Flujo actual de los procesos de información

Los procesos involucrados en la gestión de la información del colectivo de carrera en el momento actual no se realizan con medios y soportes que garanticen la seguridad de los datos, se realizan de forma manual, los documentos importantes están en formatos Word y Excel que por errores se pueden perder y no poseen seguridad alguna. Se intercambian entre los directivos por medios de reuniones y para agilizar este proceso es la presente aplicación que garantiza el almacenamiento de la información y genera de forma segura e instantánea documentos facilitando el trabajo del usuario.

#### 1.2.2 Descripción de los procesos que serán automatizados

Será objeto de automatización todo el proceso de la gestión en el Colectivo de Carrera de la Universidad de Matanzas, que incluye la gestión del Plan de Proceso Docente. Este documento se realiza cada curso por el coordinador carrera y en él se registran todas las asignaturas que va a recibir el estudiante por años dependiendo del plan de estudio correspondiente. Se automatizará también el proceso de realización de la Solicitud de Servicios, que no es más que el documento donde se recoge la distribución de asignaturas por semestre. Este documento se envía al vicedecano de la facultad correspondiente y éste lo entrega a los jefes de los departamentos de las disciplinas implicadas en la solicitud para la designación de los profesores por asignaturas. El documento que se origina de este proceso se titula Balance de carga y la presente aplicación pretende facilitar todo este proceso ya que lo genera automáticamente.

#### 1.3 Antecedentes de la investigación

En recientes búsquedas y consultas realizadas, no se obtuvo conocimiento de algún sistema similar al que se pretende implementar, no obstante algunas aplicaciones pueden ser adaptables al negocio como sigue:

**Aplicación Web para el apoyo a los ajustes de planes de estudio en la Universidad de Matanzas.** Esta aplicación tiene como principio la sistematización del proceso de ajuste de plan de estudio en la Universidad de Matanzas. Constituye un antecedente ya que tiene la información de los planes de estudio.

**Sistema de Gestión de la Nueva Universidad (SIGENU).** En el SIGENU se plasma toda la documentación asociada al plan de estudios pero no se realiza la documentación que se genera a partir del proceso de solicitud de asignaturas entre otros.

## 1.4 Metodología de la investigación

### 1.4.1 Métodos Teóricos Utilizados

**Análisis – Síntesis:** Este método se evidenció en el análisis de los elementos de la situación problemática, posibilitando descomponer mentalmente un todo complejo en sus partes y cualidades, para luego sobre la base de los resultados alcanzados previamente por el análisis establecer mentalmente la unión entre las partes, permitiendo descubrir relaciones y características generales entre los elementos de la realidad.

**Histórico – Lógico:** Permitió estudiar la trayectoria, evolución y desarrollo de los sistemas de gestión de la información en las universidades, directamente dirigidos a los colectivos de carrera.

**Inductivo – Deductivo:** Su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

### 1.4.2 Métodos Empíricos Utilizados

**Entrevista:** Este método fue esencial para el desarrollo de la investigación ya que permitió conocer las necesidades reales del cliente y los beneficios que podría aportar la aplicación.

**Análisis documental:** El análisis de documentos incluye la revisión y estudio del Reglamento Docente y Metodológico para comprender las funcionalidades del colectivo de carrera y también de los coordinadores de carrera y jefes de departamento ya que son posibles usuarios de la aplicación. También la revisión de los Planes de Estudio para obtener información, la estructura de los Planes de Proceso Docente, de la Solicitud de Servicios y del Balance de Carga.

**Observación:** La observación fue necesaria desde el comienzo para comprender como se realizaban los procesos y detectar sus insuficiencias para posteriormente darles solución.

## 1.5 Fundamentación de la metodología utilizada

Para la elaboración de la aplicación que se propone como solución al problema planteado es necesario un conjunto de técnicas y herramientas que permitan agilizar el proceso, así como una metodología de desarrollo de software que sirva de guía en el transcurso del mismo.

Las metodologías de desarrollo de software tradicionales aparecen como pesadas y poco eficientes cuando se trata de proyectos pequeños como el que ocupa esta tesis. Como respuesta a esto, se ha visto en los últimos tiempos el surgimiento de “Metodologías Ágiles”. Estos nuevos métodos buscan un punto medio entre la ausencia de procesos y el abuso de los mismos, proponiendo un proceso cuyo esfuerzo valga la pena. Los métodos ágiles cambian significativamente algunos de los énfasis de las metodologías “clásicas” (RocketTheme, 2009):

- Los métodos ágiles son adaptables en lugar de predictivos. Los métodos “clásicos” tienden a intentar planear una gran parte del proceso del software en gran detalle para un plazo largo de tiempo. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio.
- Los métodos ágiles son orientados a la gente y no orientados al proceso. El objetivo de los métodos “clásicos” es definir un proceso que funcionará bien independientemente de quien lo utilice. Los métodos ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo.

#### 1.5.1 Metodología de desarrollo de software: XP

“Extreme Programming” o “Programación Extrema” es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo. La metodología también enfatiza el trabajo en equipo. Tanto gerentes como clientes y desarrolladores son partes del mismo equipo dedicado a entregar software de calidad. (Fuentes, 2015) Extreme Programming (XP) surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. El ciclo de vida de un proyecto XP incluye, entender lo que el cliente necesita, estimar el esfuerzo, crear la solución y entregar el producto final al cliente. XP propone un ciclo de vida dinámico, donde se admite expresamente que, en muchos casos, los clientes no son capaces de especificar sus requerimientos al comienzo de un proyecto.

Por esto, se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas. XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien

parece que ésta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales. (Joskowicz, 2012)

## 1.6 Herramientas y tecnologías utilizadas

### 1.6.1 Tendencias tecnológicas a considerar

#### **Arquitectura Cliente Servidor:**

Es una arquitectura de procesamiento cooperativo de información donde uno de los componentes pide servicios a otro. Es una relación entre procesos corriendo en máquinas separadas. El cliente es un consumidor de servicios y el servidor es un proveedor de servicios e interactúan por un mecanismo de pasaje de datos: pedido de servicios – respuesta.

Cliente: Una estación de trabajo o microcomputador (PC: Computadora Personal) conectado a una red que le permite acceder y gestionar una serie de recursos. En este caso se refiere a un microcomputador conectado al sistema de información y en el que se realiza una parte mayoritaria de los procesos.

Servidor o Back-end: Está dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs y suministran servicios tales como acceso a base de datos, archivos, comunicaciones, impresión, procesamiento de imágenes, correo entre otros. Este mecanismo favorece flexibilidad y dinamismo en las organizaciones (Álvarez, 2007).

#### **Patrón de diseño Modelo Vista Controlador**

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, entre otras). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (Tedeschi, 2013):

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario, manejo de las peticiones, de la seguridad, cargar la configuración de la aplicación y realiza los cambios apropiados en el modelo o en la vista.

### **Mapeo de Objetos a Bases de datos (ORM) Doctrine**

Un ORM convierte datos entre sistemas de tipos de un lenguaje de programación orientado a objetos y una base de datos relacional. El resultado es una base de datos orientada a objetos que puede usarse desde el lenguaje de programación utilizado: las clases son tablas de la base de datos y los objetos son registros. Así resulta más fácil crear y manipular tablas y datos.

Una de las ventajas de utilizar estas capas de abstracción de objetos/relacional es que evita utilizar una sintaxis específica de un sistema de bases de datos concreto. Esta capa transforma automáticamente las llamadas a los objetos en consultas SQL optimizadas para el sistema gestor de bases de datos que se está utilizando en cada momento. De esta forma, es muy sencillo cambiar a otro sistema de bases de datos completamente diferente en mitad del desarrollo de un proyecto. Estas técnicas son útiles por ejemplo cuando se debe desarrollar un prototipo rápido de una aplicación y el cliente aún no ha decidido el sistema de bases de datos que más le conviene.

### **Sistema Gestor de Base de Datos (SGBD)**

Un Sistema Gestor de Base de Datos es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. A través de él se maneja todo acceso a la base de datos con el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

### **MySQL**

MySQL que es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido (Martínez, 2014). 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.

### **Entorno de desarrollo integrado (IDE): PhpStorm**

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.

Estas son algunas de las características claves de PhpStorm:

- ✓ Editor inteligente: Editor de código PHP inteligente con finalización para la codificación más rápida.
- ✓ Análisis consciente de PHP: Conocimiento de código PHP avanzado y navegación rápida.
- ✓ Tu laboratorio de pruebas de código: Pruebas de unidad PHP asistidas con la interfaz de usuario de ejecutor de pruebas.

#### **1.6.2 Lenguajes de programación:**

Entre los distintos lenguajes de programación para la Web que existen en la actualidad, se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente/Servidor, característica de los sistemas Web.

- El primer grupo lo integran los lenguajes que se ejecutan en el cliente, ejemplos de ellos son: HTML, CSS, JavaScript, estos se encargan de aportar dinamismo a la aplicación en los navegadores.
- El segundo grupo está formado por los lenguajes que se ejecutan en el servidor, dentro de los que se encuentra ASP, PHP y Java, estos lenguajes se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a bases de datos y el tratamiento de la información.

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. El navegador es una especie de aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.

Del lado del cliente:

**HTML 5** (HyperText Markup Language)

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido). Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. La utilización de HTML como uno de los lenguajes para el desarrollo, responde a las necesidades de manipulación y maquetación de los elementos visuales de la aplicación. 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 (Álvarez, 2012).

### **CSS 3 (Cascade Style Sheets)**

Hojas de estilo en cascada (o CSS, siglas en inglés de Cascading Stylesheets) no es como tal un lenguaje de programación, sino un lenguaje de estilos, con el cual se define la apariencia de una página web definiendo como debe mostrarse cada uno de sus elementos HTML. CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento. (Cristian, 2008)

### **TypeScript 2.6.2**

TypeScript es un lenguaje de programación de distribución y código abierto desarrollado por Microsoft. Su propio fabricante lo define como un superset (superconjunto) de JavaScript que, en esencia, mejora JavaScript añadiéndole funciones de tipado y de programación orientada a objetos. El propósito de TypeScript es cubrir las carencias o deficiencias del lenguaje JavaScript, haciéndolo un lenguaje más completo, escalable, seguro e incluso más fácil de utilizar. Una de sus principales características frente al JavaScript es la programación orientada a objetos basado en clases, lo cual, aunque de alguna manera era posible con JavaScript, resultaba complicado ya que el lenguaje no está pensado para ello. Últimamente este lenguaje está ganando popularidad ya que el potente framework Angular está totalmente desarrollado con este lenguaje y ellos mismos recomiendan utilizar TypeScript para desarrollar en Angular.

Del lado del servidor:

## **PHP (Hypertext Pre-Processor)**

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy. (Achour, 2007)

Ventajas:

- ✓ Es un lenguaje multiplataforma.
- ✓ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ✓ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.

### **1.6.3 Framework de desarrollo web**

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

#### **Symfony 3.4**

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

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



- Fácil de instalar y configurar en la mayoría de plataformas
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- 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. (Potencier, y otros, 2011)

### **Bootstrap 4.0.0**

Es un framework web de código abierto desarrollado por Twitter. Contiene plantillas para casi todo, como plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS y JavaScript. La utilización de este framework, ahorra la parte de tener que escribir tu propio código CSS u hoja de estilos, ya que ya viene todo montado y tan solo tienes que añadir ciertas clases a tus elementos HTML para que la página cobre vida como por arte de magia. Sus posibilidades son ilimitadas (Solanas, y otros).

### **Angular 5.2.0**

El desarrollar aplicaciones web hoy en día, se ha popularizado tanto que todos los desarrolladores están tratando de crear Aplicaciones modernas en la web, y para tal objetivo se necesitan herramientas modernas, una de esas herramientas es Angular 5, Angular es el framework para JavaScript estándar para crear webs. Es uno de los frameworks más populares para desarrollar aplicaciones modernas y escalables en el lado del cliente: Angular 2, Angular 4 y Angular 5 han dado un salto de calidad con respecto a las versiones anteriores del framework, como AngularJS (Bastidas, 2017).

### **1.7 Conclusiones del capítulo**

En el desarrollo del primer capítulo se dio a conocer las bases teóricas sobre las cuales se sustenta la propuesta de trabajo. Se analiza cómo se efectúa el proceso y se corroboró la necesidad de diseñar esta aplicación para solucionar los problemas existentes. Se realiza un estudio detallado de las tendencias existentes para el desarrollo de software y se definen las tecnologías y técnicas que se aplicarán al diseño

e implementación de la aplicación informática. Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología de programación extrema XP. Se realiza un análisis detallado de las tecnologías a utilizar optando por utilizar el lenguaje de programación PHP usando como entorno de desarrollo PhpStorm y el gestor de bases de datos MySQL.

## **Capítulo 2: Descripción de la solución propuesta.**

### **2.1 Introducción al capítulo**

En el capítulo se exponen los elementos necesarios para la descripción de la solución propuesta. A través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Se aplica la Metodología XP, Extreme Programming (Programación Extrema), metodología ágil de desarrollo, con el objetivo de garantizar el diseño de un programa lo más ajustado posible y se logra como ventaja la incorporación del cliente como un miembro del equipo de desarrollo.

### **2.2 Descripción de la solución**

Se propone desarrollar una aplicación web que sea capaz de gestionar la información del Colectivo de Carrera de la Universidad de Matanzas de forma tal que los datos existentes puedan ser almacenados sin riesgo de que puedan perderse o deteriorarse y sea fácil consultarlos. Les permitirá a los coordinadores de carrera conformar el Plan de Proceso Docente correspondiente a cada cohorte estudiantil con mayor facilidad ya que ese documento estará automatizado, al igual que otros documentos como la Solicitud de Servicios y los Jefes de Departamentos podrán realizar el Balance de Carga de una manera más agilizada. Con el desarrollo de este software la información va a estar más organizada y segura.

### **2.3 Etapa de planificación**

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando "Historias de usuarios", las que sustituyen a los tradicionales "casos de uso". Una vez obtenidas las "Historias de Usuarios", los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene "riesgos" que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una

vez acordado este cronograma, comienza una fase de iteraciones, en donde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios” (Joskowicz, 2012).

### 2.3.1 Equipo de trabajo y roles

La metodología XP define roles de trabajo asociando a cada uno con diversas actividades. A continuación, se definen los roles, quedando designado el programador que sería el encargado de producir el código del sistema, el jefe de proyecto y el cliente que no es más que el que escribe las historias de usuario, les asigna la prioridad y diseña las pruebas funcionales para validar su implementación. A continuación, se muestra la asignación de estos roles a las personas responsables (Joskowicz, 2012).

Miembros	Roles
Lorena Varela García	Programador
Walfredo González Hernández	Jefe del Proyecto, Tester,
Leticia Fuentes	Cliente

*Tabla 1 Equipo de trabajo y roles*

### 2.3.2 Historias de Usuario Iniciales

Las “Historias de usuarios” sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario (Fuentes, 2015).

#### **Escalas equivalentes a la prioridad en el negocio:**

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

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

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

**Escala Nominal de Riesgo en Desarrollo:**

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

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

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

A continuación se colocan los requisitos funcionales detectados:

No	Nombre	Prioridad	Riesgo	Esfuerzo	Iteración	Entrega
1	Diseño y Creación de la base de datos	Media	Alto	1	1	1
2	Diseño de la interfaz de usuario.	Media	Medio	1		
3	Autenticación del Usuario	Alta	Medio	1		
4	Gestionar usuario	Alta	Alto	2		
5	Gestionar Facultad	Alta	Alto	2	2	2
6	Gestionar Carrera	Alta	Alto	2		
7	Gestionar Departamento	Alta	Alto	2		
8	Gestionar Disciplina	Alta	Alto	2	3	3
9	Gestionar Asignatura	Alta	Alto	2		
10	Gestionar Profesor	Alta	Alto	2		
11	Gestionar Cohorte	Alta	Alto	2		
12	Gestionar Plan de Proceso Docente PPD	Alta	Alto	3	4	4
13	Generar PPD	Media	Medio	1		
14	Generar Solicitud de Servicios	Media	Medio	2		

15	Generar Balance de Carga	Media	Medio	2		
----	--------------------------	-------	-------	---	--	--

*Tabla 2 Planificación de las Historias de Usuario*

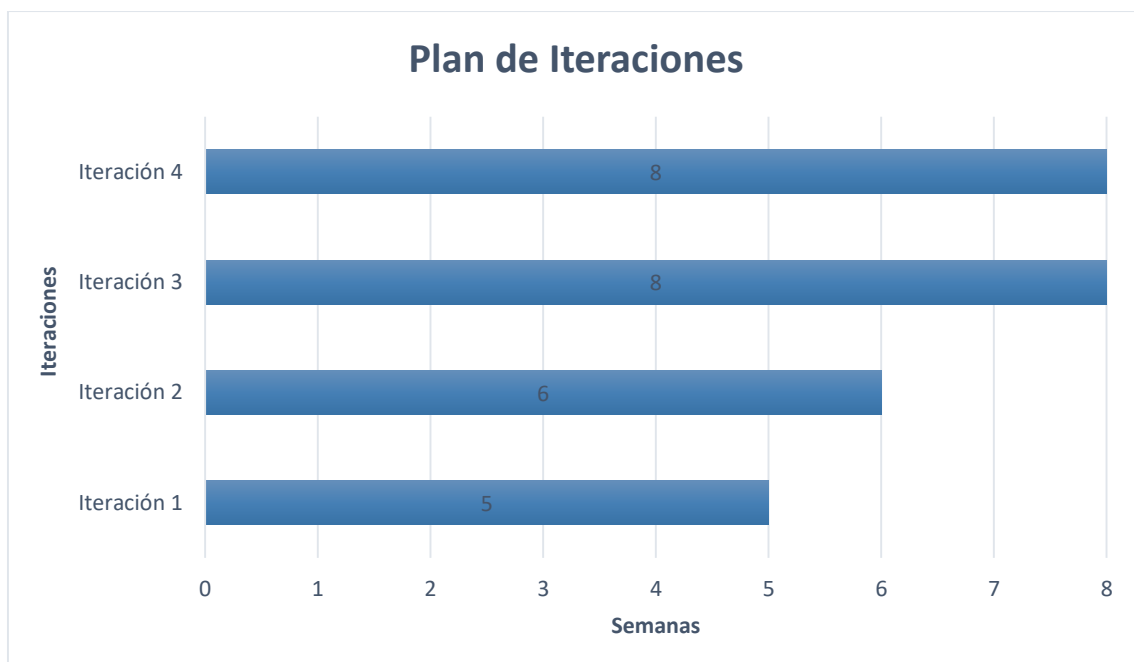
Una vez detectadas los requisitos pues es importante elaborar el plan de entrega y de iteraciones.

### 2.3.3 Plan de Iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación.

El proyecto fue dividido en cuatro iteraciones, por lo que se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones se tuvo en cuenta la opinión del cliente a través de las entrevistas que se le realizaron continuamente antes de comenzar a desarrollar cada iteración donde se tomaron todos los acuerdos necesarios. Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades.

En la ilustración se muestran las cuatro iteraciones y las semanas en las que se desarrollan



### Ilustración 1 Plan de Iteraciones

Posterior al plan de iteraciones se realiza la estimación del proyecto. En este caso se utilizó puntos de función para tener una estimación temprana.

#### 2.3.4 Análisis del costo

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 asociados (Pressman, 2010).

La estimación del costo de un software es el proceso de predecir la cantidad de esfuerzo requerido para 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, Boton – Up, Top – Down, entre otros.

#### Método Puntos de Función

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

#### Componentes:

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

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

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

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

**EIF:** Grupos de datos que se mantienen externamente.

Una vez obtenidos los diferentes elementos del sistema se utilizan las tablas reflejadas en el **Anexo 4** para asignar pesos en función del número de atributos que tengan y el número de archivos a los que afecte.

Componentes	Cantidad de Componentes por su Peso	Total
EI (Entradas)	9 * 4	32
EO (Salidas)	12 * 4	48
EQ (Consultas)	12 * 4	48
FLI	14 * 5	70

FLE	0*5	0
-----	-----	---

### **Cálculo de los Puntos de Función sin Ajustar (PFSA)**

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

$$\text{PFSA} = 32 + 48 + 48 + 70 = 198$$

### **Cálculo de los Puntos de Función Ajustados (PFA)**

$$\text{PFA} = \text{PFSA} * [0.65 + (0.01 * \text{ACT})]$$

### **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 5**. Cuando cada Factor tenga un valor, se suman todos y así obtenemos el ACT. A continuación se muestra dicho procedimiento.

#### **Cálculo del factor de ajuste.**

$$\text{PFA} = \text{PFSA} * [0.65 + [0.01 * \text{ACT}]]$$

$$\text{PFA} = 198.12 * [0.65 + (0.01 * 29)]$$

$$\text{PFA} = 198.12 * [0.65 + 0.29]$$

$$\text{PFA} = 198.12 * 0.94$$

$$\text{PFA} = 186.12$$

#### **Cálculo del Esfuerzo**

**Líneas de Código (LC) = PFA + Líneas por PF**

$$\text{LC} = 186.12 * 20$$

$$\text{Líneas de Código} = 3722.4$$

#### **Esfuerzo en horas / persona (E)**

$$\text{E} = \text{PFA} / [1/8 \text{ persona-hora}]$$

$$\text{E} = 186.12 / (1/8)$$

$$\text{E} = 186.12 * 8$$

$$\text{E} = 1488 / 192 \text{ ctd de horas al mes.}$$

**E = 7.75 meses E = 8 meses**

### **Cálculo del Presupuesto del Proyecto**

Costo Total del proyecto = sueldo de 1 participante \* cantidad de participantes\*

Tiempo de desarrollo

Costo Total del proyecto = 400 \* 1 \* 8

Costo Total del proyecto = 3200

#### **2.3.5 Análisis de los beneficios**

Realizando un análisis del costo respecto a los beneficios descritos a lo largo de este trabajo se puede concluir que los aspectos positivos son superiores a los costos. Además, debe señalarse que para esta institución este software resultó gratuito puesto que es el resultado del trabajo de diploma del autor.

#### **2.3.6 Plan de Entregas**

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

<b>Entrega</b>	<b>Descripción</b>
<b>1</b>	Esta entrega comprende <ul style="list-style-type: none"><li>- Diseño y creación de la base de datos</li><li>- Diseño de la interfaz de usuario</li><li>- Autenticación</li><li>- Gestión de usuario</li></ul>
<b>2</b>	Esta entrega comprende <ul style="list-style-type: none"><li>- Gestión de Facultades</li><li>- Gestión de Carreras</li><li>- Gestión de Departamentos</li></ul>



<b>3</b>	Esta entrega comprende <ul style="list-style-type: none"> <li>- Gestión de Disciplinas</li> <li>- Gestión de Asignaturas</li> <li>- Gestión de Profesores</li> <li>- Gestión de Cohortes</li> </ul>
<b>4</b>	Esta entrega comprende <ul style="list-style-type: none"> <li>- Gestión de PPD</li> <li>- Generar PPD</li> <li>- Generar de Solicitudes de Servicio</li> <li>- Generar de Balance de Carga</li> </ul>

Tabla 3 Plan de entrega

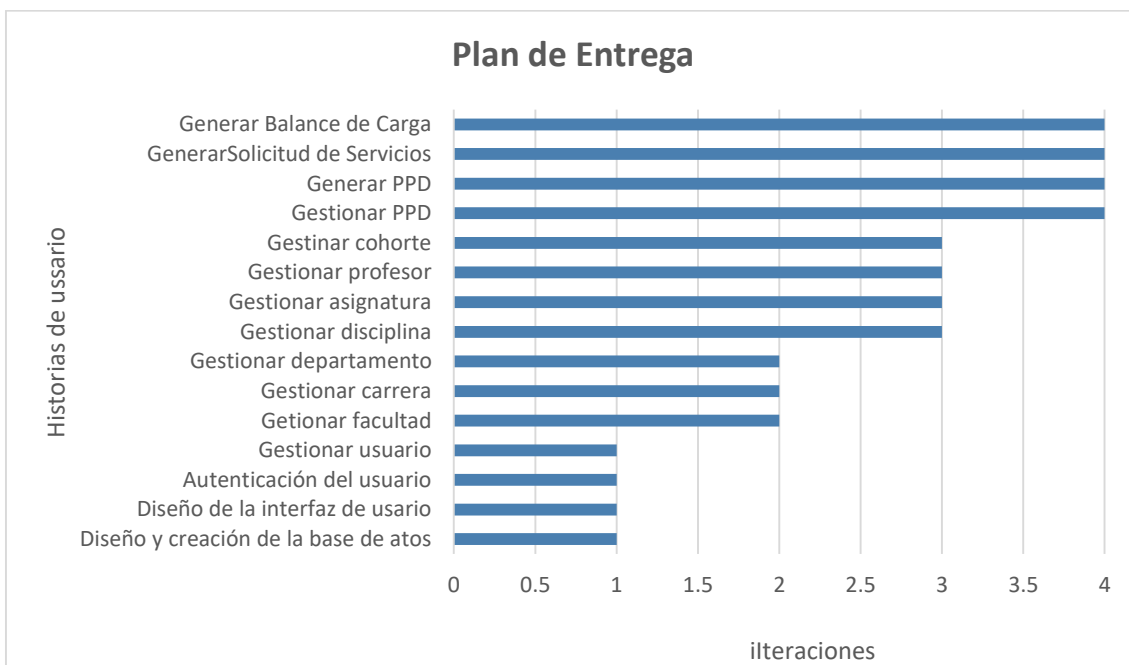


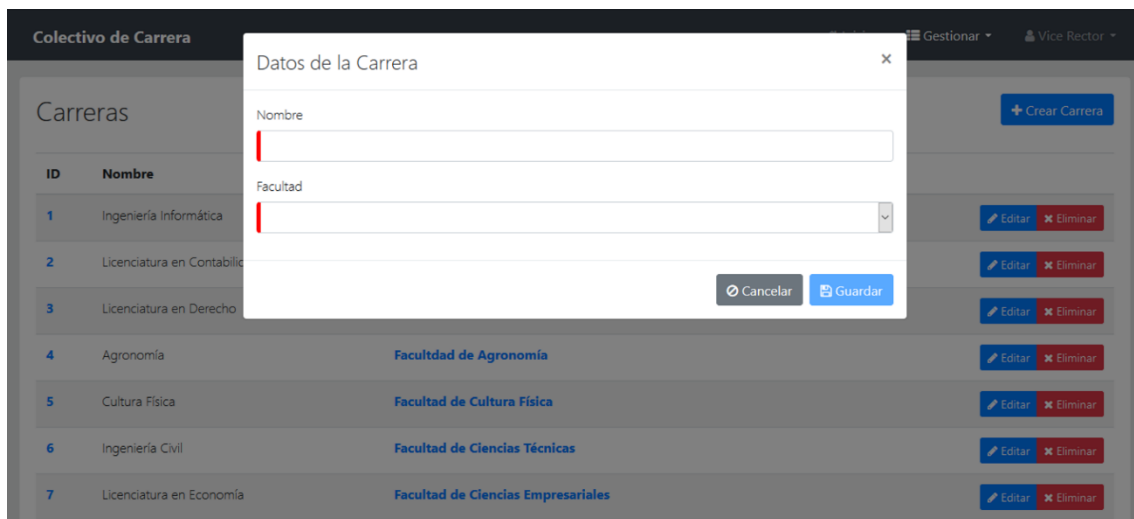
Ilustración 2 Plan de entregas

## 2.4 Etapa de diseño

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar. Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y reutilización del código. Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema (Joskowicz, 2012).

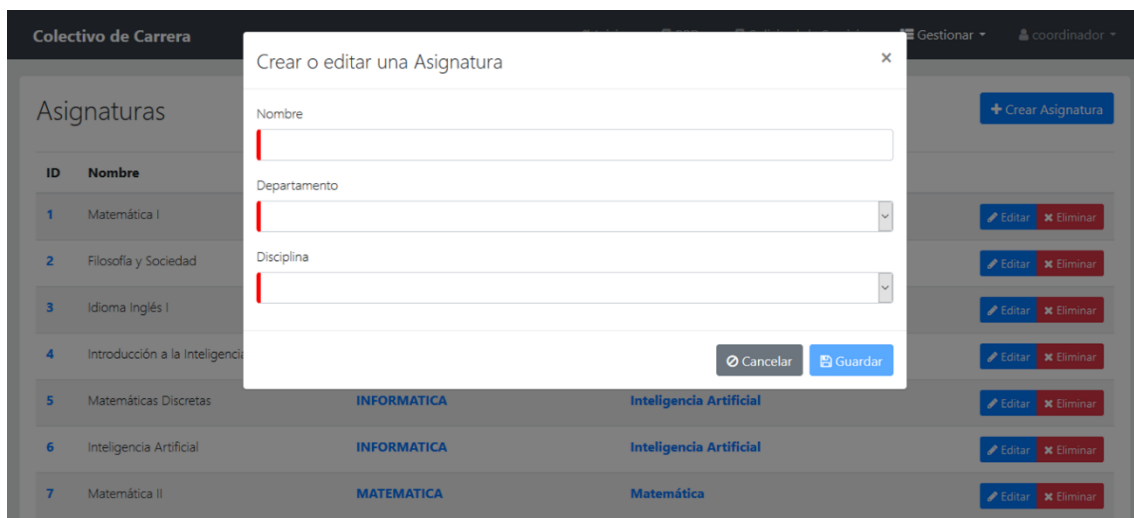
### 2.4.1 Prototipo de la interfaz de usuario

Se muestran ilustraciones de la interfaz de usuario. En la ilustración se muestra la interfaz que permite al usuario insertar una carrera.



*Ilustración 3 Insertar carrera*

En la ilustración se muestra a interfaz que permite al usuario crear una asignatura



*Ilustración 4 Insertar Asignatura*

En la ilustración se muestra a interfaz que permite al usuario crear plan de proceso docente.

Colectivo de Carrera Inicio PPD Solicitud de Servicio Gestionar coordinador

Nuevo Plan de Proceso Docente Carrera: Ingeniería Informática

Plan de Estudio Tipo de Curso Curso

Asignaturas

Nombre Horas(Semanales/Totales) Currículo Evaluacion Semestre

Guardar

### Ilustración 5 Crear Plan de Proceso Docente

A continuación se muestran las historias de usuarios que se consideran más importante en detalle para que se pueda comprender el proceso:

Historia de Usuario	
<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre:</b> Diseño y creación de la Base de datos	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración Asignada:</b> 1
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Se diseña e implementa la base de datos en el gestor MySQL	
<b>Observaciones:</b> Verificar si a la base de datos se le estableció una contraseña.	

Tabla 4 Diseño y creación de la Base de datos

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Administrador, Coordinador JfeDpto, Vicerrector
<b>Nombre:</b> Autenticación del Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración Asignada:</b> 1
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> El usuario escribe su nombre y contraseña para acceder al sistema, de ser correctos se le mostrará un entorno de trabajo acorde con su nivel de acceso y de no serlo, el sistema mostrará un error.	
<b>Observaciones:</b> Se debe verificar si son correctos los datos para acceder al sistema y si el usuario está registrado en las bases de datos	

Tabla 5 Autenticación del Usuario

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gestionar Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 1
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el administrador accede a esta funcionalidad para gestionar los usuarios. Podrá insertar, editar y eliminar un usuario del sistema además de asignarle un rol.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 6 Gestionar Usuarios*

Historia de Usuario	
<b>Número:</b> 5	<b>Usuario:</b> Vicerrector
<b>Nombre:</b> Gestionar Facultad	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para gestionar una facultad. El usuario podrá insertar, editar y eliminar una facultad del sistema.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos sean correctos.	

*Tabla 7 Gestionar Facultad*

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gestionar Carrera	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para gestionar un carrera. El usuario podrá insertar, editar y eliminar una carrera del sistema.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 8 Gestionar Carrera*

Historia de Usuario	
<b>Número:</b> 7	<b>Usuario:</b> Administrador

<b>Nombre:</b> Gestionar Departamento	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para gestionar un departamento. El usuario podrá insertar, editar y eliminar un departamento del sistema.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 9 Gestionar Departamento*

Historia de Usuario	
<b>Número:</b> 8	<b>Usuario:</b> Coordinador
<b>Nombre:</b> Gestionar Disciplina	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para gestionar una disciplina. El usuario podrá insertar, editar y eliminar una disciplina del sistema.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 10 Gestionar Disciplina*

Historia de Usuario	
<b>Número:</b> 9	<b>Usuario:</b> Coordinador
<b>Nombre:</b> Gestionar Asignatura	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para gestionar una asignatura. El usuario podrá insertar, editar y eliminar una asignatura del sistema.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 11 Gestionar Disciplina*

Historia de Usuario	
<b>Número:</b> 10	<b>Usuario:</b> JefeDpto
<b>Nombre:</b> Gestionar Profesor	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto

<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para gestionar un profesor. El usuario podrá insertar, editar y eliminar un profesor del sistema.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 12 Gestionar Profesor*

<b>Historia de Usuario</b>	
<b>Número:</b> 11	<b>Usuario:</b> Coordinador, Vicerrector
<b>Nombre:</b> Gestionar Cohorte	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el coordinador accede a esta funcionalidad para gestionar una cohorte. El usuario podrá insertar, editar y eliminar una cohorte del sistema y asignarle un Plan de Proceso Docente.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 13 Gestionar Cohorte*

<b>Historia de Usuario</b>	
<b>Número:</b> 12	<b>Usuario:</b> Coordinador
<b>Nombre:</b> Gestionar PPD	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración Asignada:</b> 4
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el coordinador accede a esta funcionalidad para gestionar un PPD. El usuario podrá insertar, editar y deshabilitar un PPD	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

*Tabla 14 Gestionar Plan de Proceso Docente PPD*

<b>Historia de Usuario</b>	
<b>Número:</b> 13	<b>Usuario:</b> Coordinador
<b>Nombre:</b> Generar PPD	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración Asignada:</b> 4

<b>Programador responsable:</b> Lorena Varela García
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para generar un PPD. Se exportará en formato Excel.

Tabla 15 Generar Plan de Proceso Docente PPD

Historia de Usuario	
<b>Número:</b> 14	<b>Usuario:</b> Coordinador
<b>Nombre:</b> Generar Solicitud de Servicios	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración Asignada:</b> 4
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el usuario accede a esta funcionalidad para generar una Solicitud de Servicios. Se exportará en formato Excel.	
<b>Observaciones:</b>	

Tabla 16 Generar Solicitud de Servicios

Historia de Usuario	
<b>Número:</b> 15	<b>Usuario:</b> JefeDpto
<b>Nombre:</b> Generar Balance de Carga	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración Asignada:</b> 4
<b>Programador responsable:</b> Lorena Varela García	
<b>Descripción:</b> Inicia cuando el jefe de departamento accede a esta funcionalidad para generar un Balance de Carga. Se exportará en formato Excel.	
<b>Observaciones:</b> Se debe verificar que los datos introducidos en cada uno de los campos sean los correctos.	

Tabla 17 Generar Balance de Carga

Es importante descomponer cada una de las historias de usuario en tareas de ingeniería. Cada una de ellas se muestra a continuación:

#### 2.4.2 Tareas de Ingeniería

No	Nombre HU	No	Tarea de Ingeniería
1	Diseño y creación de la base de datos.	1	Diseño de la base de datos
		2	Creación de la base de datos
2	Diseño de la interfaz de usuario	3	Interfaz de usuario
3	Autenticación del usuario	4	Autenticación del usuario
		5	Insertar usuario
		6	Modificar usuario

<b>4</b>	Gestionar usuario	7	Eliminar usuario
		8	Listar usuario
<b>5</b>	Gestionar facultad	9	Insertar facultad
		10	Modificar facultad
		11	Eliminar facultad
		12	Listar facultad
<b>6</b>	Gestionar carrera	13	Insertar carrera
		14	Modificar carrera
		15	Eliminar carrera
		16	Listar carrera
<b>7</b>	Gestionar departamento	17	Insertar departamento
		18	Modificar departamento
		19	Eliminar departamento
		20	Listar departamento
<b>8</b>	Gestionar disciplina	21	Insertar disciplina
		22	Modificar disciplina
		23	Eliminar disciplina
		24	Detalles disciplina
		25	Listar disciplina
<b>9</b>	Gestionar asignatura	26	Insertar asignatura
		27	Modificar asignatura
		28	Eliminar asignatura
		29	Listar asignatura
<b>10</b>	Gestionar profesor	30	Insertar profesor
		31	Modificar profesor
		32	Eliminar profesor
		33	Listar profesor
<b>11</b>	Gestionar cohorte	34	Insertar cohorte
		35	Modificar cohorte
		36	Eliminar cohorte
		37	Listar cohorte
		38	Asignar PPD
<b>12</b>	Gestionar Plan de Proceso Docente	39	Insertar PPD
		40	Modificar PPD
		41	Deshabilitar PPD



		42	Detalles PPD
		43	Listar PPD
13	Generar Plan de Proceso Docente	44	Generar Plan de Proceso Docente
14	Generar solicitud de servicios	45	Generar solicitud de servicios
15	Generar Balance de carga	46	Generar Balance de carga

*Tabla 18 Resumen de las tareas de ingeniería.*

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

Tarea de Ingeniería	
<b>Número de Tarea:</b> 1	<b>Número de Historia de Usuario:</b> 1
<b>Nombre:</b> Diseño de la base de datos	
<b>Tipo de tarea:</b> Diseño	<b>Puntos estimados:</b> 0.5
<b>Programador Responsable:</b> Lorena Varela García	
<b>Descripción:</b> Analizar profundamente el negocio para crear la base de datos que permita almacenar dicha información.	

*Tabla 19 TI: Diseño de la base de datos*

Tarea de Ingeniería	
<b>Número de Tarea:</b> 2	<b>Número de Historia de Usuario:</b> 1
<b>Nombre:</b> Creación de la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Programador Responsable:</b> Lorena Varela García	
<b>Descripción:</b> Crear la base de datos con sus respectivas relaciones y la integridad correspondiente entre las tablas.	

*Tabla 20 TI: Creación de la base de datos*

Tarea de Ingeniería	
<b>Número de Tarea:</b> 5	<b>Número de Historia de Usuario:</b> 4
<b>Nombre:</b> Insertar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Programador Responsable:</b> Lorena Varela García	
<b>Descripción:</b> Se mostrará un formulario donde el usuario con los permisos necesarios podrá insertar un usuario y podrá asignarle el rol correspondiente	

Tabla 21 TI: Insertar usuario

Tarea de Ingeniería	
<b>Número de Tarea:</b> 6	<b>Número de Historia de Usuario:</b> 4
<b>Nombre:</b> Modificar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Programador Responsable:</b> Lorena Varela García	
<b>Descripción:</b> Por cada usuario que se muestre en el listado de usuarios, habrá la opción de modificar, esta mostrará un formulario con los datos del usuario seleccionado donde se podrán modificar, todos los campos deben estar llenos.	

Tabla 22 TI: Modificar usuario

Tarea de Ingeniería	
<b>Número de Tarea:</b> 7	<b>Número de Historia de Usuario:</b> 4
<b>Nombre:</b> Eliminar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Programador Responsable:</b> Lorena Varela García	
<b>Descripción:</b> Por cada usuario que se muestre en el listado de usuarios, habrá la opción de eliminar, se eliminará de la base de datos los datos relacionados con ese usuario.	

Tabla 23 TI: Eliminar usuario

Tarea de Ingeniería	
<b>Número de Tarea:</b> 46	<b>Número de Historia de Usuario:</b> 12
<b>Nombre:</b> Insertar PPD	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Programador Responsable:</b> Lorena Varela García	
<b>Descripción:</b> Se mostrará un formulario donde el usuario con los permisos necesarios podrá insertar un PPD. Está compuesto por datos como nombre, tipo de curso, curso. El usuario seleccionará las asignaturas que le corresponden al plan y a cada una de ellas le asignará horas semanales y totales, currículo, evaluación y semestre.	

Tabla 24 TI: Insertar Plan de Proceso Docente

Tarea de Ingeniería	
<b>Número de Tarea:</b> 47	<b>Número de Historia de Usuario:</b> 12
<b>Nombre:</b> Modificar PPD	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Programador Responsable:</b> Lorena Varela García	

**Descripción:** Se mostrará un formulario donde el usuario con los permisos necesarios podrá modificar los datos de un PPD.

Tabla 25 TI: Modificar Plan de Proceso Docente

Tarea de Ingeniería	
Número de Tarea: 47	Número de Historia de Usuario: 12
Nombre: Deshabilitar PPD	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador Responsable: Lorena Varela García	
Descripción: El usuario con los permisos necesarios podrá deshabilitar PPD. El plan deshabilitado no podrá ser asignado a una cohorte.	

Tabla 26 TI: Deshabilitar Plan de Proceso Docente

### 2.4.3 Modelo físico de la Base de datos

En la ilustración se observa el modelo físico de la base de datos de la aplicación web.

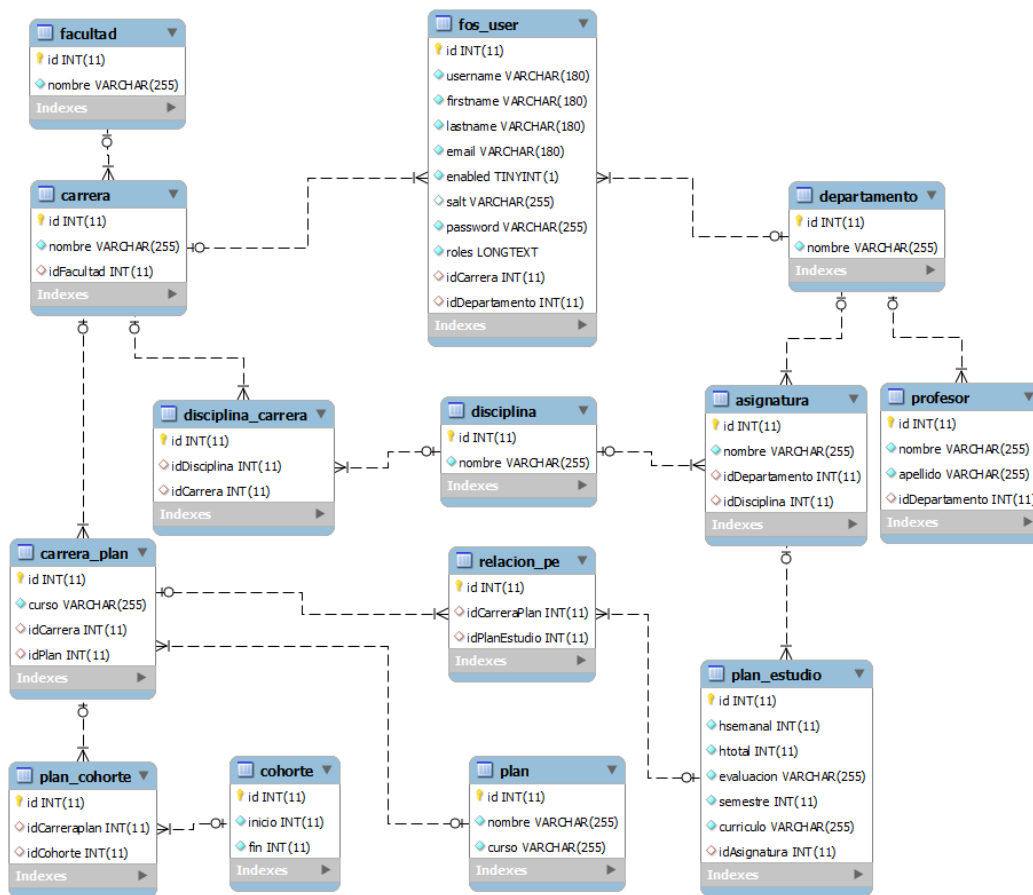


Ilustración 6 Modelo de la base de datos

### 2.4.4 Tarjetas de Clase, Responsabilidad y Colaboración

Se muestran algunas tarjetas de Clase, Responsabilidad y Colaboración (CRC) creadas

usuario	
<b>Superclase:</b>	
<b>Subclase:</b>	
<b>Descripción:</b> En esta clase se guardan los elementos referidos a los usuarios.	
Atributos	
idusuario	int
usuario	String
nombre	String
apellido	String
correo	String
rol	String

*Tabla 27 Tarjeta CRC Usuario*

## 2.5 Conclusiones parciales del Capítulo

En este capítulo se plantean las etapas necesarias para desarrollar el software según la metodología XP, con la excepción de las pruebas. Se escriben las historias de usuario tareas iniciales que se agrupan en iteraciones y entregan. Se pudo concretar final de las iteraciones la entrega del sistema informático completamente en funcionamiento.

## 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 desarrollo 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.

### 3.2 Pruebas al software

El proceso de pruebas es el instrumento más adecuado para determinar el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar 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

Según (Loaiza, et al., 2010), el plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento.

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 28** el plan de pruebas

No.	Historia de Usuario	Pruebas a realizar
1	Diseño y Creación de la base de datos.	Test base de datos
2	Diseño de la interfaz de usuario.	Test diseño de la interfaz de usuario

<b>3</b>	Autenticación del usuario	Test Autenticación del usuario
<b>4</b>	Gestionar usuario	Test Insertar usuario. Test Eliminar usuario. Test Editar usuario. Test Listar usuario.
<b>5</b>	Gestionar facultad	Test Insertar facultad. Test Eliminar facultad. Test Editar facultad. Test Listar facultad.
<b>6</b>	Gestionar carrera	Test Insertar carrera. Test Eliminar carrera. Test Editar carrera. Test Listar carrera.
<b>7</b>	Gestionar departamento	Test Insertar departamento. Test Eliminar departamento. Test Editar departamento. Test Listar departamento.
<b>8</b>	Gestionar disciplina	Test Insertar disciplina. Test Eliminar disciplina. Test Editar disciplina. Test Listar disciplina. Test Destalles disciplina
<b>9</b>	Gestionar asignatura	Test Insertar asignatura. Test Eliminar asignatura. Test Editar asignatura. Test Listar asignatura.
<b>10</b>	Gestionar profesor	Test Insertar profesor. Test Eliminar profesor. Test Editar profesor. Test Listar profesor.
<b>11</b>	Gestionar cohorte	Test Insertar usuario. Test Eliminar usuario. Test Editar usuario. Test Asignar PPD.
<b>12</b>	Gestionar Plan de Proceso Docente	Test Insertar PPD. Test Deshabilitar PPD.

		Test Editar PPD. Test Listar PPD.
13	Generar Plan de Proceso Docente	Test Generar Plan de Proceso Docente
14	Generar Solicitud de Servicios	Test Generar Solicitud de Servicios
15	Generar Balance de Carga	Test Generar Balance de Carga

Tabla 28 Plan de Pruebas

### 3.2.2 Pruebas de aceptación

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. (Loaiza, y otros, 2010)

Caso de Prueba	
Número: 1	Historia de usuario: 1
Nombre de caso de prueba: Test Base de Datos	
Descripción: Verifica el funcionamiento de la base de datos.	
Condición de ejecución: Estar conectado a la base de datos.	
Entradas: Valores para leer o escribir en la base de datos por ejemplo: usuario, contraseña.	
Resultado esperado: Se muestran o guardan los datos correctamente.	
Evaluación: Prueba Satisfactoria.	

Tabla 29 PA Test Base de datos

Caso de Prueba	
Número: 3	Historia de usuario: 3
Nombre de caso de prueba: Test Autenticación del usuario	
Descripción: Se inserta el nombre de usuario y la contraseña para entrar al sistema. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación. Luego se insertarán los datos de manera correcta para comprobar esta funcionalidad.	

<b>Condición de ejecución:</b> Que la aplicación web este en ejecución
<b>Entradas:</b> Insertar datos: nombre de usuario y contraseña
<b>Resultado esperado:</b> El sistema debe alertar al usuario cuando se inserten datos erróneos o cuando se dejen campos en blanco. Cuando se inserten los datos correctamente, según el rol que tenga el usuario, se le otorgan los permisos correspondientes.
<b>Evaluación:</b> Prueba Satisfactoria.

Tabla 30: PA Test Autenticación del usuario

Caso de Prueba	
<b>Número:</b> 9	<b>Historia de usuario:</b> 9
<b>Nombre de caso de prueba:</b> Test Gestionar asignatura	
<b>Descripción:</b> Se inserta el nombre de la asignatura para crear una nueva asignatura, se inserta de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados. Se modifican los datos de la asignatura, se modifican incorrectamente dejando campos en blanco para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados. Se elimina la asignatura aceptando el mensaje de confirmación.	
<b>Condición de ejecución:</b> Estar conectado a la base de datos.	
<b>Entradas:</b> Datos de la asignatura	
<b>Resultado esperado:</b> El sistema debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos.	
<b>Evaluación:</b> Prueba Satisfactoria.	

Tabla 31: PA Test Gestionar asignatura

En la **Tabla 32** se muestran las clases de equivalencia de las pruebas de aceptación al sistema en la HU: Gestionar asignatura

Condición de entrada	Clases válidas	Representante	Clases inválidas	Representante
nombre	1. Cualquier cadena de caracteres alfabéticos que comiencen con	Simulación	2. Cadena que contenga caracteres numéricos	Simulación



	mayúscula. Permite espacio.		3.Cadena que no comience con mayúscula	simulación
			4-Cadena vacía.	NULL
<b>disciplina</b>	5.seleccionar algunas de las opciones	Matemática Aplicada	6.no seleccionar ninguna de las opciones	NULL
<b>departamento</b>	7. seleccionar algunas de las opciones	Matemática	8.no seleccionar ninguna de las opciones	NULL

Tabla 32: Clases de equivalencia Insertar asignatura

En la **Tabla 33** se muestra un resumen de algunos casos de pruebas necesarios para realizar las pruebas de aceptación.

Resumen de Casos de prueba

No	Clases equiv.	nombre	disciplina	departamento	Resultado
1	1,5,7	Simulación	Matemática Aplicada	Matemática	Asignatura insertada
2	1,5,8	Simulación	Matemática	Null	Inserte un departamento
3	1,6,7	Simulación	Null	Matemática	Inserte una disciplina
4	1,6,8	Simulación	Null	Null	Botón crear deshabilitado. NO se puede insertar la asignatura
5	2,5,7	Filosof4a	Marxismo	Marxismo	Escriba solo letras
6	2,5,8	Filosof4a	Marxismo	Null	Escriba solo letras. Botón crear deshabilitado. NO se

					puede insertar la asignatura
<b>7</b>	2,6,7	Filosof4a	Null	Marxismo	Escriba solo letras. Seleccione una disciplina
<b>8</b>	2,6,8	Filosof4a	Null	Null	Escriba solo letras. Botón crear deshabilitado. NO se puede insertar la asignatura
<b>9</b>	3,5,7	filosofía	Marxismo	Marxismo	Comience la escritura con mayúscula.
<b>10</b>	3,5,8	filosofía	Marxismo	Null	Comience la escritura con mayúscula. Seleccione un departamento
<b>11</b>	3,6,7	filosofía	Null	Marxismo	Comience la escritura con mayúscula. Seleccione una asignatura
<b>12</b>	3,6,8	filosofía	Null	Null	Comience la escritura con mayúscula. Botón crear deshabilitado. NO se puede insertar la asignatura
<b>13</b>	4,5,7	Null	Español	Español	El nombre es obligatorio.
<b>14</b>	4,5,8	Null	Español	Null	El nombre es obligatorio. Seleccione un departamento
<b>15</b>	4,6,7	Null	Null	Español	El nombre es obligatorio. Seleccione una asignatura
<b>16</b>	4,6,8	Null	Null	Null	El nombre es obligatorio. Botón crear deshabilitado. NO se puede insertar la asignatura

Tabla 33: Resumen de Casos de prueba

A continuación se muestran algunas Tablas de Prueba por Caso de Prueba

Test insertar asignatura

Tabla de Prueba	
<b>No.Caso de Prueba</b>	1
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de insertar un equipo en el sistema (Para cubrir las clases válidas 1,5,7)
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Pentium - Disco Duro de 1T - Memoria RAM de 4GB.
<b>Software</b>	Sistema Operativo Windows 7 o versiones posteriores – Base de Datos MySQL – Navegador de internet Mozilla Firefox 3.5.
<b>Personal</b>	Encargados de Pruebas
<b>Caso de Prueba</b>	
<b>Datos de Entrada</b>	Nombre: Simulación, disciplina: Matemática Aplicada, departamento: Matemática
<b>Resultados Esperados</b>	Asignatura insertada correctamente.
<b>Resultados Obtenidos</b>	SI(X) NO()
<b>Casos de Excepción</b>	
<b>Aprobado por: Walfredo Gonzáles Hernández</b>	<b>Cago:</b> Jefe del Proyecto, Tester <b>Líder:</b> Lorena Varela

Tabla 34 Test Insertar asignatura. Caso de preba 1

27	Redes de Computadoras	INFORMATICA	Infraestructura de Sistemas Informáticos	<a href="#">Editar</a> <a href="#">Eliminar</a>
28	Ingeniería de Software I	INFORMATICA	Ingeniería y Gestión de Software	<a href="#">Editar</a> <a href="#">Eliminar</a>
29	Optativa III	INFORMATICA	Optativas	<a href="#">Editar</a> <a href="#">Eliminar</a>
30	Simulación	MATEMATICA	Matemática Aplicada	<a href="#">Editar</a> <a href="#">Eliminar</a>
31	Optativa IV	INFORMATICA	Optativas	<a href="#">Editar</a> <a href="#">Eliminar</a>
32	Optativa V	INFORMATICA	Optativas	<a href="#">Editar</a> <a href="#">Eliminar</a>
34	Optativa VIII	INFORMATICA	Optativas	<a href="#">Editar</a> <a href="#">Eliminar</a>
35	Trabajo de Diploma	INFORMATICA	Práctica Profesional	<a href="#">Editar</a> <a href="#">Eliminar</a>

Ilustración 7 Test Insertar asignatura. Caso de preba 1

Tabla de Prueba	
<b>No.Caso de Prueba</b>	4
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de insertar un equipo en el sistema (Para cubrir las clases 1,6,8)
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Pentium - Disco Duro de 1T - Memoria RAM de 4GB.
<b>Software</b>	Sistema Operativo Windows 7 o versiones posteriores – Base de Datos MySQL – Navegador de internet Mozilla Firefox 3.5.
<b>Personal</b>	Encargados de Pruebas
<b>Caso de Prueba</b>	
<b>Datos de Entrada</b>	Nombre: Español, disciplina: Null, departamento: Null
<b>Resultados Esperados</b>	Botón guardar queda desactivado. No se puede insertar la asignatura.
<b>Resultados Obtenidos</b>	SI(X) NO()
<b>Casos de Excepción</b>	
<b>Aprobado por: Walfredo Gonzáles Hernández</b>	<b>Cago:</b> Jefe del Proyecto, Tester <b>Líder:</b> Lorena Varela

Tabla 35 Test Insertar asignatura. Caso de preba4



Ilustración 8 Test Insertar asignatura. Caso de preba4

Tabla de Prueba	
<b>No.Caso de Prueba</b>	13
<b>Requerimiento</b>	Estar conectado a la base de datos.
<b>Objetivo</b>	Probar la acción de insertar un equipo en el sistema (Para cubrir las clases 4,5,7)
<b>Tipo de Prueba</b>	Funcional.
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Pentium - Disco Duro de 1T - Memoria RAM de 4GB.
<b>Software</b>	Sistema Operativo Windows 7 o versiones posteriores – Base de Datos MySQL – Navegador de internet Mozilla Firefox 3.5.
<b>Personal</b>	Encargados de Pruebas
<b>Caso de Prueba</b>	
<b>Datos de Entrada</b>	Nombre: Null, disciplina: Español, departamento: Español.
<b>Resultados Esperados</b>	El nombre es requerido
<b>Resultados Obtenidos</b>	SI(X) NO()
<b>Casos de Excepción</b>	
<b>Aprobado por: Walfredo Gonzáles Hernández</b>	<b>Cago:</b> Jefe del Proyecto, Tester <b>Líder:</b> Lorena Varela

Tabla 36 Insertar asignatura. Caso de prueba 13

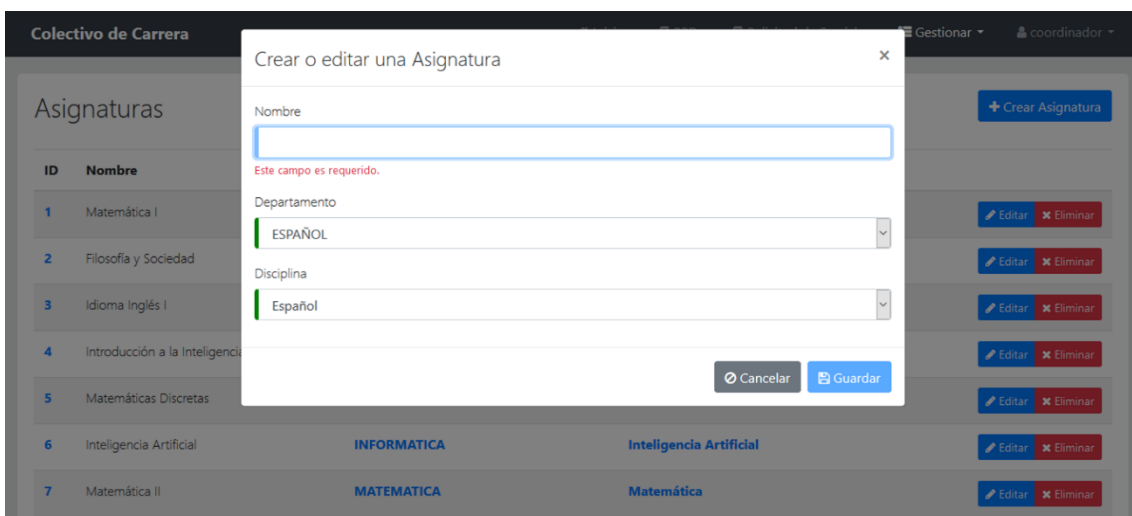


Ilustración 9 Insertar asignatura. Caso de prueba 13

### 3.2.3 Pruebas de seguridad

Las pruebas de seguridad buscan medir la Confidencialidad, Integridad y Disponibilidad de los datos, desde la perspectiva del aplicativo, es decir partiendo a identificar amenazas y riesgos desde el uso o interface de usuario final. Una vez ejecutadas las pruebas de seguridad es posible medir y cuantificar los riesgos a los cuales se ven expuestos los aplicativos tanto en la infraestructura interna como externa (V&V Quality, 2016)

De acuerdo como se plantea en (Seguridad para todos, 2012), VEGA es una plataforma de software libre diseñada para realizar y ejecutar pruebas sobre la seguridad de las páginas web. Las principales características son:

- Capacidad de realización de Análisis de Vulnerabilidades
- Ejecución de pruebas de Accesibilidad.
- Análisis del contenido
- Mensajes de notificación personalizables
- Modelo de datos propio

En la ilustración se muestra el resultado obtenido.

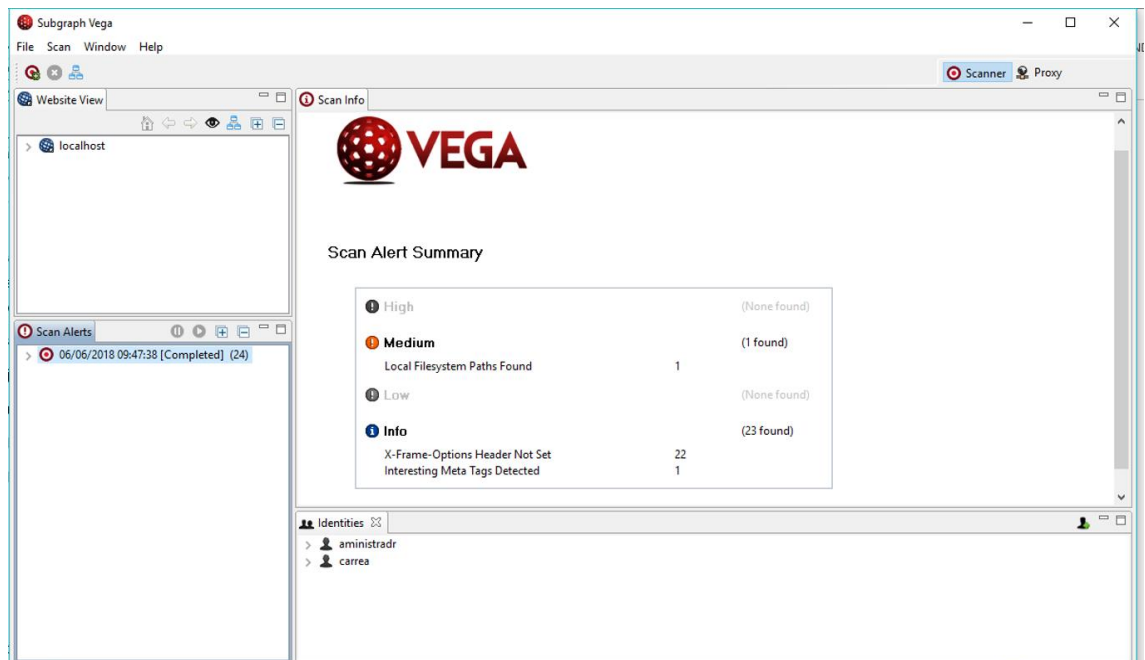


Ilustración 10 Prueba de seguridad. VEGA

### 3.3 Análisis de los resultados obtenidos

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las pruebas realizadas, se solucionaron varios problemas que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión.

Las primeras pruebas fueron planeadas y ejecutadas en módulos individuales del programa y a medida que fueron avanzando se desplazaron a módulos integrados, hasta que finalmente llegaron al sistema completo y se logró obtener un software cuyas funciones se encuentra en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento. El desarrollo del sistema cumple las expectativas trazadas al inicio del proyecto y satisface al cliente en su totalidad.

### 3.4 Conclusiones parciales

Las pruebas realizadas permitieron validar el funcionamiento de la aplicación y los resultados satisfactorios de dichas pruebas. Una vez realizadas las pruebas se logró brindarle al cliente una versión del software que facilitara gestión de la información en el Colectivo de Carrera de la Universidad de Matanzas.

## Conclusiones Generales

1. Se elaboró un mapa de procesos asociado al campo de investigación a partir del análisis de los trabajos más importantes relacionados con la gestión de la información del Colectivo de Carrera de la Universidad de Matanzas
2. Se utilizó la metodología XP para el desarrollo de la aplicación Web que se presenta, así como tecnologías de avanzada para el diseño de la aplicación.
3. Se implementa la solución utilizando el lenguaje de programación PHP que permite agilizar los procesos.
4. Mediante la aplicación de pruebas a la aplicación Web, resultó posible obtener resultados favorables, con el consiguiente análisis de errores detectados que fueron subsanados como parte del desarrollo de esta aplicación.



## 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:

- Extender la aplicación a las demás Universidades del país.
- Aprovechar las posibilidades de información almacenadas en la aplicación e implementar funcionalidades que respondan a nuevos intereses de la universidad.
- Agregarle el módulo de Programa de la asignatura, donde se detalle el contenido de cada una de ellas facilitando la labor del profesor.

## Bibliografía

**Achour, M. 2007.** *Manual de PHP*. 2007.

**Álvarez, Miguel Ángel. 2007.** DesarrolloWeb. [En línea] 30 de 8 de 2007.  
<http://desarrolloweb.com>.

**Álvarez, R. 2012.** Introducción al HTML. [En línea] 2012.  
<http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>..

**Bastidas, william. 2017.** Angular 5: Inicio rápido con angular-cli y angular material. [En línea] 2017. <https://blog.ng-classroom.com/blog/angular/inicio-rapido-angular-material/>.

**Bastidas, William.** <https://blog.faztweb.com/2017/04/angular-complete.html>.  
[En línea] <https://blog.faztweb.com/2017/04/angular-complete.html>.

**Cristian. 2008.** Ventajas de usar CSS. [En línea] 2008.  
<http://www.stardustxs.com/2008/03/05/ventajas-de-usar-css/>.

**Fernández, P. 2011.** *JetBrains anuncia PHP Storm*. 2011.

**Fuentes, J.R.L. 2015.** *Desarrollo de Software Ágil. Extreme Programming y Scrum*. 2015.

<http://www.tic.siteal.iipe.unesco.org/politicas/1102/programa-rector-de-la-informatizacion-de-la-sociedad-cubana>. [En línea]  
<http://www.tic.siteal.iipe.unesco.org/politicas/1102/programa-rector-de-la-informatizacion-de-la-sociedad-cubana>.

**Joskowicz, Ing. José . 2012.** *Reglas y Prácticas en Extreme Programming*. 2012.

**Loaiza, Vanesa y Zorro, Laura. 2010.** *Plan de Pruebas de Software*. 2010.

**Martínez, D. 2014.** PostgreSQL vs MySQL. . [En línea] 2014.  
<Http://danielpecos.com/documents/postgresql-vs-mysql/>.

**Meléndez, Karin, Abraham, Dávila y Carla, Basurto. 2005.** Estimación del Esfuerzo en la etapa Post-Arquitectura usando el modelo COCOMO II Versión 2.0. 30 de 01 de 2005.

**Milán Licea, Dra María Rosa , Ruiz Echevarría, MSc. Hilda y Fraga Guerra, MSc. Elena. 2017.** *El colectivo de carrera: su labor de asesoría.* s.l. : Congreso Uiversidad, 2017.

**Potencier, Fabien y Zaninotto, François . 2011.** *Symfony, guía definitiva.* 2011.

**Pressman. 2010.** *Ingeniería del software.Un enfoque práctico.* Mexico : s.n., 2010.

**Risueño, P. 2013.** *Comenzando con Bootstrap, framework responsive.* 2013.

**RocketTheme. 2009.** *Metodologías de desarrollo de software.* 2009.

**Solanas, Antonio y Sierra, Vicenta.** *Bootstrap: fundamentos e introducción a sus aplicaciones.*

**Tedeschi, N. 2013.** *¿Qué es un Patrón de Diseño?* 2013.

**Vela Valdés, Dr Juan y ValdésAsán, Lic Jorge. 2007.** *RESOLUCIÓN No. 210/07.* 2007.

## Anexos

### Anexo 1 Malla Curricular del Plan de Proceso Docente

| MS-11C

#### Plan del Proceso Docente

Carrera: \_\_\_\_\_ Tipo de curso: \_\_\_\_\_

Plan de Estudio \_\_\_\_ vigente a partir del curso escolar \_\_\_\_\_

Asignatura	h/c	EF
1er Año. 1er Semestre		
2do Año. 1er Semestre		
3er Año. 1er Semestre		
4to Año. 1er Semestre		
5to Año. 1er Semestre		
6to Año. 1er Semestre		

Asignatura	h/c	EF
1er Año. 2do Semestre		
2do Año. 2do Semestre		
3er Año. 2do Semestre		
4to Año. 2do Semestre		
5to Año. 2do Semestre		
6to Año. 2do Semestre		

## Anexo 2 Formato Solicitud de Servicios

	A	B	C	D	E	F	G
1	<b>CARRERA INGENIERÍA INFORMÁTICA</b>						
2	<b>SERVICIOS DE ASIGNATURAS PARA EL PRIMER SEMESTRE DEL CURSO 2015-2016</b>						
3	<b>CURSO POR ENCUENTROS</b>						
4	<b>PRIMER AÑO PRIMER SEMESTRE</b>						
5	<b>ASIGNATURA</b>	<b>H/S</b>	<b>H T</b>	<b>Examen Final</b>	<b>DEPARTAMENTO</b>	<b>No Grupos</b>	<b>Semanas</b>
6	Matemática I	3	48	X	Matemática	1	16
7	Algebra Lineal y Geometría Analítica	3	48	X Según documento de reunión nacional de CNC y dictamen recibido de cambios de CPE (22-06-2011)	Matemática	1	16
8	Introducción a la Informática	2	32	X	Informática	1	16
9							
10	<b>SEGUNDO AÑO PRIMER SEMESTRE</b>						
11	<b>ASIGNATURA</b>	<b>H/S</b>	<b>H T</b>	<b>Examen Final</b>	<b>DEPARTAMENTO</b>	<b>No Grupos</b>	<b>Semanas</b>
12	Introducción a la Programación	3	48	X	Informática	1	16
13	Matemática III	3	48	X	Matemática	1	16
14	Probabilidades y Estadística	2	32	X	Matemática	1	16
15	Economía Política del capitalismo y de la Construcción del Socialismo	2	32	X	Marxismo Leninismo	1	16
16							
17	<b>TERCER AÑO PRIMER SEMESTRE</b>						
18	<b>ASIGNATURA</b>	<b>H/S</b>	<b>H T</b>	<b>Examen Final</b>	<b>DEPARTAMENTO</b>	<b>No Grupos</b>	<b>Semanas</b>
19	Defensa y Seguridad Nacional	2	32	X	Preparación para la Defe	1	16
20	Introducción a la Inteligencia Artificial	2	32	X	Informática	1	16
21	Estructura de Datos I	3	48	X	Informática	1	16
22	Problemas Sociales de la Ciencia y la Tecnología	1	16	X	Marxismo-Leninismo	1	16
23							
24	<b>CUARTO AÑO PRIMER SEMESTRE</b>						
25	<b>ASIGNATURA</b>	<b>H/S</b>	<b>H T</b>	<b>Examen Final</b>	<b>DEPARTAMENTO</b>	<b>No Grupos</b>	<b>Semanas</b>
26	Base de Datos	3	48	X	Informática	1	16
27	Gestión económica financiera	2	32	X	Contabilidad y Finanzas	1	16
28	Sistemas Operativos	2	32	X	Informática	1	16
29	Optativa 1	1	16	X		1	16
30	Lenguajes de Programación				Informática	1	16
31	Gestión de Software				Informática	1	16
32	Seguridad Informática			X	Dirección de Informatización	1	16

### Anexo 3 Formato Balance de Carga

	B	C	D	E	F	G	H	I	J	K
1	Añ	TC	Asignatura	Disciplina	Hora	Grup	Eval	Sem	Total Hoi	Profesor
21	4	CD	Optativa III Desarrollo de aplicaciones en Angular	Infraestructura de SI	42	1		18	42	Santiago
22	4	CD	Optativa III Desarrollo Ágil para la WEB (XETID)	Ing. y Gestión SW	42	1		18	42	XETID
23	4	CD	Optativa IV Seguridad Informática	Infraestructura de SI	42	1		18	42	Claritza
24	4	CD	Optativa IV Bases de Datos Avanzadas (XETID)	Infraestructura de SI	42	1		18	42	XETID
25	5	CD	Seminario Profesional	Práctica Profesional	42	1	TC	18	42	Briseida
26	5	CD	Componente Profesional del Trabajo de Curso de Seminario Profesional	Práctica Profesional	112	1		18	112	Briseida
27	5	CD	Optativa VIII Medición en Proyectos de Software	Ing. y Gestión SW	42	1		18	42	Walfredo
28	5	CD	Optativa VIII Desarrollo de Interfaces de Usuario para la Web	Ing. y Gestión SW	42	1		18	42	XETID
29	5	CD	Optativa IX Redes Avanzadas	Infraestructura de SI	42	1		18	XETID	Angel(Redes)
30	5	CD	Optativa IX Desarrollo ágil para la Web. Marco de trabajo Zeolides	Ing. y Gestión SW	42	1		18	42	XETID
31	5	CD	Optativa X: Java Serverface	Ing. y Gestión SW	42	1		18	42	Emilio
32	5	CD	Optativa X Base de Datos Avanzada	Ing. y Gestión SW	42	1		18	42	XETID
33	5	CD	Pedagogía y Didáctica de la Educación Superior		32	1		18	32	Norka(P)
34	1	CPE	Introducción a la Informática	Ing. y Gestión SW	28	1	EF	16	28	Orlenys
35	2	CPE	Introducción a la Programación	Ing. y Gestión SW	48	1	EF	16	48	Felix
36	3	CPE	Introducción a la Inteligencia Artificial	Inteligencia Artificial	32	1	EF	16	32	Orlenys
37	3	CPE	Estructura de Datos I	Ing. y Gestión SW	48	1	EF	16	48	René
38	4	CPE	Bases de Datos	Ing. y Gestión SW	48	1	EF	16	48	Angel G
39	4	CPE	Sistemas Operativos	Infraestructura de SI	32	1	EF	16	32	Penton
40	4	CPE	Optativa I Seguridad Informática	Ing. y Gestión SW	16	1	EF	16	16	Angel G

**Anexo 4** Tablas de ponderaciones para EI, EQ, EO, ILF, EIF

<b>CLASIFICACION DE ENTRADAS Y CONSULTAS</b>	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

<b>CLASIFICACION DE SALIDAS</b>	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

<b>FICHEROS LÓGICOS INTERNOS</b>	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
--	----------	---------	---------

<b>FICHEROS LÓGICOS EXTERNOS</b>	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



## Anexo 5 Cálculo del Factor de Ajuste

<b>Nº</b>	<b>Nombre del Factor</b>	<b>Valor</b>
1	Comunicación de datos	2
2	Procesamiento distribuido	2
3	Rendimiento	3
4	Configuración de Explotación Compartida	2
5	Tasa de transacciones	1
6	Entrada de Datos en Línea	2
7	Eficiencia con el Usuario Final	4
8	Actualizaciones en Línea	0
9	Procesamiento complejo	2
10	Reusabilidad del Código	1
11	Facilidad de implementación	2
12	Facilidad de Operación	4
13	Instalaciones Múltiples	2
14	Facilidad de Cambios	2
	Ajuste de Complejidad Técnica(ACT)	29

**Anexo 6** Tabla para obtener las Líneas por Puntos de Función

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