



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

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

**Aplicación web para el proceso de encuesta sobre la  
satisfacción de la calidad de los servicios en la Universidad de  
Matanzas sede "Camilo Cienfuegos"**

**Autor:** Leandro García Benítez

**Tutor:** Ing. Odalis Acosta Imbet

Msc. Yadrián Pulido García

**Matanzas, 2018**

## **Frase**

*"Ser culto es el único modo  
de ser libre"*

*José Martí*

# **Dedicatoria:**

A mi mamá Dulce Benítez García, mi papá  
Conrado García Bernal, mi hermano Leodán  
García Benítez.

# **Agradecimientos:**

Les agradezco a todas las personas que de una manera u otra formaron parte de este trabajo, a mis compañeros de aula, a los profesores, a mis tutores y un especial agradecimiento a mi familia.

## Resumen

En la Universidad de Matanzas sede "Camilo Cienfuegos" las encuestas se realizan en modelos de papel, razón por la cual se incurre en demoras, y resulta demasiado engorroso llegar a una estadística final ya que el procesamiento y almacenamiento de las mismas se hace muy difícil, por lo que no se solucionan efectivamente todos los problemas. El departamento de calidad de la universidad, ha solicitado un sistema informático para contribuir a la eficiencia del proceso de encuesta de satisfacción de la calidad de los servicios. Se ha propuesto desarrollar una aplicación web, dadas las deficiencias de la universidad y la concurrencia de un personal variado en esta. Se seguirá la arquitectura cliente - servidor. Además, se utilizará el *framework* de desarrollo Symfony y la metodología de desarrollo de software Programación Extrema; XP por sus siglas en inglés (Extreme Programming). Este trabajo dará como resultado una aplicación capaz de realizar y controlar las encuestas y a la vez dar un nivel de satisfacción de la Universidad de Matanzas.

## **Summary**

In the University of Matanzas, "Camilo Cienfuegos", the surveys are carried out on paper models, which is why delays are incurred, and it is too cumbersome to arrive at a final statistic since the processing and storage of them becomes very difficult, which does not effectively solve all the problems. The department of quality of the university, has requested a computer system to contribute to the efficiency of the process of satisfaction survey of the quality of the services. He has been proposed to develop a web application, given the deficiencies of the university and the concurrence of a varied staff in it. The client-server architecture will be followed. In addition, the Symfony development framework and the Software Development methodology Extreme Programming will be used. This work will result in an application capable of conducting and monitoring the surveys and at the same time give a level of satisfaction of the university of Matanzas.

# Índice

## Contenido

Introducción .....	1
1 Capítulos I Marco Teórico-Referencial .....	8
<b>1.1 Introducción.</b> .....	8
<b>1.2 Antecedentes del trabajo</b> .....	8
<b>1.3 Objeto de estudio.</b> .....	9
1.3.1 <b>Descripción de los conceptos y procesos fundamentales a emplear.</b> .....	10
1.3.2 <b>Flujo actual del proceso involucrado en el campo de acción.</b> 12	
1.3.3 <b>Análisis crítico de la ejecución actual de este proceso y las causas que originan la situación problemática, así como sus consecuencias.</b> .....	12
<b>1.4 Tendencias tecnológicas.</b> .....	13
1.4.1 <b>Fundamentación de la metodología utilizada.</b> .....	13
1.4.2 <b>Arquitectura cliente-servidor</b> .....	15
1.4.3 <b>Lenguajes de programación</b> .....	18
1.4.4 <b>Sistema de gestión de bases de datos</b> .....	23
1.4.5 <b>Servidor web</b> .....	25
<b>1.5 Conclusiones parciales del capítulo.</b> .....	25
2 Capítulo II Análisis, diseño y construcción de la solución propuesta	27
<b>2.1 Introducción</b> .....	27
2.1.1 <b>Etapas de Planificación.</b> .....	28
2.1.2 <b>Equipo de trabajo y roles.</b> .....	28
2.1.3 <b>Historias de Usuario iniciales.</b> .....	29

2.1.4	<b>Plan de iteraciones</b> .....	34
2.1.5	<b>Reuniones</b> .....	35
2.1.6	<b>Plan de entregas</b> .....	35
2.1.7	<b>Beneficios tangibles e intangibles</b> .....	36
2.1.8	<b>Metáforas</b> .....	36
<b>2.2</b>	<b>Etapa de diseño.</b> .....	<b>37</b>
2.2.1	<b>Prototipo de interfaz de usuario</b> .....	37
2.2.2	<b>Tareas a desarrollar</b> .....	38
2.2.3	<b>Diseño de la base de datos.</b> .....	44
2.3	<b>Conclusiones parciales del capítulo.</b> .....	45
3	Capítulo 3: Validación de la solución propuesta.....	46
<b>3.1</b>	<b>Introducción</b> .....	<b>46</b>
<b>3.2</b>	<b>Pruebas al software</b> .....	<b>46</b>
3.2.1	<b>Plan de pruebas</b> .....	46
3.2.2	<b>Pruebas de aceptación</b> .....	47
<b>3.3</b>	<b>Análisis de los resultados.</b> .....	<b>54</b>
3.4	<b>Beneficios tangibles e intangibles. Estimación del costo.</b> 55	
<b>3.5</b>	<b>Conclusiones del Capítulo</b> .....	<b>56</b>
4	Conclusiones generales.....	57
5	Recomendaciones .....	58
6	Bibliografía .....	59

# Índice de tablas

Tabla 1 Equipo de trabajo y roles .....	29
Tabla 2 Historias de Usuarios iniciales .....	31
Tabla 3 HU 1.....	31
Tabla 4 HU 2.....	32
Tabla 5 HU 3.....	32
Tabla 6 HU 4.....	33
Tabla 7 HU 5.....	33
Tabla 8 HU 6.....	34
Tabla 9Tareas de Iteración (Elaboración propia) .....	38
Tabla 10 TI 1 .....	39
Tabla 11 TI 5 .....	39
Tabla 12 TI 7 .....	39
Tabla 13 TI 9 .....	40
Tabla 14 TI 10 .....	40
Tabla 15 TI 11 .....	40
Tabla 16 TI 12 .....	41
Tabla 17 TI 13 .....	41
Tabla 18 TI 14 .....	41
Tabla 19 TI 15 .....	42
Tabla 20 TI 16 .....	42
Tabla 21 TI 18 .....	42
Tabla 22 TI 19 .....	43
Tabla 23 TI 20 .....	43
Tabla 24 Plan de pruebas .....	47
Tabla 25: Plan de Pruebas .....	47
Tabla 26 pruebas de aceptación .....	48
Tabla 27 Prueba de aceptación No.2 .....	48
Tabla 28 Prueba de aceptación No.3 .....	49
Tabla 29 Prueba de aceptación No.4 .....	50
Tabla 30 Prueba de aceptación No.5 .....	51

Tabla 31 Prueba de aceptación No.6 .....	52
Tabla 32 Prueba de aceptación No.7 .....	53
Tabla 33 Prueba de aceptación No.16.....	53
Tabla 34 Prueba de aceptación No. 17.....	54

# Índice de ilustraciones

Ilustración 1 Plan de iteraciones .....	35
Ilustración 2 Planificación de las Iteraciones en semanas (Elaboración propia) .....	36
Ilustración 3 Interfaz principal.....	37
Ilustración 4 Base de datos.....	44
Ilustración 5 Autenticarse .....	49
Ilustración 6 crear encuesta .....	50
Ilustración 7 eliminar encuesta.....	51
Ilustración 8 publicar encuesta .....	52

## Introducción

Desde el surgimiento de la informática el objetivo era facilitar operaciones que manualmente eran muy agobiantes de realizar, en ocasiones por la complejidad de ellas y en otras por la masiva repetición que se necesitaba para conseguir el fin añorado.

La informatización nació con el propósito de eliminar las complejidades de procesos que necesitan transformar, conservar, crear, eliminar o transmitir gran cantidad de información. Antes de esta era muy difícil almacenar datos estadísticos de ciertos campos como del económico o del social debido a la aglomeración de información y llegar a resultados de operaciones complejas en corto período de tiempo, era casi imposible.

Desde finales del siglo XX se ha visto un auge en el desarrollo de software de gestión en varias ramas del desarrollo socio-económico como cabe mencionar SAP, que es el más utilizado a nivel mundial y pionero en este mundo, un 24% de los usuarios apuestan por este programa de gestión integral; también otros como ORACLE, SAGE, INFOR, entre muchos otros de igual o mayor uso. (autores)

El desarrollo de software a nivel mundial está en alza, su mercado se rige por dos vertientes de acuerdo a ofertas en dos planos de intersección: el plano horizontal, que indica la forma en que se entrega el software y el plano vertical, que clasifica al mismo, de acuerdo a su fin o área de utilización. Entre las herramientas más utilizadas para el desarrollo de estos se encuentran los diagramas UML, los cuales permiten mediante un diseño abstracto, los desarrolladores y el cliente definan el sistema, su funcionamiento y sus funcionalidades; otra de las herramientas son las del lenguaje en el que se vaya a trabajar ya sea .Net de Microsoft, C o PHP. Para el diseño se encuentran las Plataformas IDE y las Herramientas de diseño gráfico, para el desarrollo de páginas web están **Bluefish, Amaya, Coda 2, Microsoft Expressions Web** o **Adobe Dreamweaver** entre otras.

La creación de estos ha contribuido mucho al desarrollo de la ciencia y la tecnología ya que las ha surtido de información valiosa para ciertas investigaciones, les ha permitido archivar resultados, analizarlos y compararlos con otras indagaciones y así llegar a conclusiones más relevantes. Los software han logrado ampliar el alcance de estas ramas de desarrollo pudiendo hacer, observar y buscar cosas que para los humanos se torna de difícil acceso.

En Cuba a partir del siglo XXI se trabaja por desarrollar software con calidad y con eficiencia; muestra de ello es la fundación de la Universidad de las Ciencias Informáticas (UCI) por el Comandante en Jefe Fidel Castro, para encaminar la informatización de la comunidad cubana y formar profesionales capaces de convertirse en el eje de la industria del software cubano, creándose en dicho centro aplicaciones informáticas para campos de la esfera educacional.

De esta realidad no están exenta el resto de las universidades cubanas como la Universidad de La Habana(UH) o la sede universitaria de Matanzas "Camilo Cienfuegos", pues en ella se trabajan con sistemas informáticos capacitados para perfeccionar y agilizar sus procesos educativos como por ejemplo el de ingreso de nuevo personal a la institución, de desarrollo de catálogos y materiales de cursos, de garantizar la disponibilidad de recursos humanos y materiales necesarios para alcanzar los objetivos, de desarrollar, revisar y actualizar los planes de estudios entre otro muchos.

En la sede universitaria "Camilo Cienfuegos" de Matanzas el proceso de encuesta de satisfacción de la calidad de los servicios se lleva a cabo de forma manual, habitual con modelos de papel, no siendo este el método más idóneo para realizar estos procedimientos, puesto que se dificulta el procesamiento, almacenamiento y control de los datos estadísticos.

Existen demoras en el proceso de encuestas y no se controla la información debido a la forma de guardado y la sistematicidad con se

chequean estos aspectos. Todo ello dificulta la obtención de datos estadísticos por lo que se imposibilita dar solución a los aspectos negativos resultantes. Actualmente las encuestas se hacen en hojas de papel, lo que provoca una sobrecarga de trabajo para el responsable de realizar dicha tarea; lo anterior planteado expresa la **situación problemática**.

Todo esto trae consigo el siguiente **problema científico**, ¿cómo automatizar el proceso de encuesta sobre la satisfacción de la calidad de los servicios en la Universidad de Matanzas sede "Camilo Cienfuegos"?

El **objetivo general** es desarrollar un sistema informático para el proceso de encuesta sobre la satisfacción de la calidad de los servicios en la Universidad de Matanzas sede "Camilo Cienfuegos".

Se propone para resolver el problema la siguiente **hipótesis**: si se desarrolla un sistema informático para el proceso de encuesta sobre la satisfacción de la calidad de los servicios entonces se agilizará y automatizará esta gestión.

El **objeto de estudio** del presente trabajo de tesis es el proceso de encuesta sobre la de satisfacción de la calidad de los servicios en la Universidad de Matanzas sede "Camilo Cienfuegos" y el **campo de acción** la automatización del proceso de encuesta sobre la satisfacción de la calidad de los servicios.

Los **objetivos específicos** son:

- Sistematizar los referentes teóricos del proceso de encuesta de satisfacción de la calidad de los servicios.
- Analizar metodologías y herramientas para el desarrollo del sistema informático del proceso de encuesta de satisfacción de la calidad de los servicios.
- Implementar el sistema informático para el proceso de encuesta de satisfacción de la calidad de los servicios en la sede "Camilo Cienfuegos" de la Universidad de Matanzas.

- Validar la propuesta implementada con las pruebas de la metodología de desarrollo y con el criterio del cliente.

Las **tareas de investigación** son:

- Sistematización de los referentes teóricos del proceso de encuesta de satisfacción de la calidad de los servicios.
- Análisis de las metodologías y herramientas para el desarrollo del sistema informático proceso de encuesta de satisfacción de la calidad de los servicios.
- Implementación del sistema informático para el proceso de encuesta de satisfacción de la calidad de los servicios en la sede "Camilo Cienfuegos" de la Universidad de Matanzas.
- Validación de la propuesta implementada con las pruebas de la metodología de desarrollo y con el criterio del cliente.

Para el desarrollo del software se utilizaron diversos **métodos y técnicas** como **métodos teóricos** de análisis histórico – lógico, de análisis y síntesis e inductivo - deductivo. Para aplicar los **métodos empíricos**, se utilizaron técnicas de observación, entrevistas y análisis de documentos.

Se propone utilizar la metodología de desarrollo de software *Extreme Programinig* (XP) la cual es de un uso fácil y según las características de este proyecto es la más adecuada debido a que requiere de pocos desarrolladores, no es demasiada extensa y se guía por pruebas que validan su funcionamiento.

Se aprecian como **beneficio** la creación de una herramienta automatizada para el proceso de encuestas de satisfacción de la calidad de los servicios y de reportes de fácil interpretación para la toma de decisiones de los directivos de la universidad y así se acelera el proceso de encuestas sobre la calidad en la Universidad de Matanzas sede "Camilo Cienfuegos" lo que facilita la obtención de resultados.

Con la realización de este software se proporciona un instrumento de gran **aporte práctico** como la creación de la herramienta informática

para el proceso de encuestas sobre la calidad de la universidad de Matanzas "Camilo Cienfuegos".

Como **resultados esperados** se quiere que la Universidad de Matanzas sede "Camilo Cienfuegos" cuente con una herramienta informática capaz de perfeccionar el proceso de encuesta de satisfacción de la calidad de los servicios.

### **Metodología de la investigación empleada.**

#### **Métodos teóricos empleados.**

- **El método de análisis histórico y el lógico:** permitió estudiar la trayectoria y desarrollo del proceso de encuestas existentes, así como sus deficiencias en la Universidad de Matanzas sede "Camilo Cienfuegos".
  
- **Los métodos de análisis y de síntesis:** este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división mental del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.
  
- **Inducción-deducción:** su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

#### **Métodos empíricos empleados.**

Para utilizar los métodos empíricos se ponen en práctica las siguientes técnicas:

- **La observación científica:** acompañó la investigación desde los primeros momentos, a través de la cual se conoció el estado y proceder de todo el proceso de encuesta sobre la satisfacción de la calidad de los servicios, y se obtuvo la información primaria acerca de los objetos investigados.
- **La entrevista:** aportó datos esenciales a la investigación puesto que el entrevistado es la persona que propuso el desarrollo del sistema en primer lugar. Fue útil en distintos momentos de la investigación; fundamentalmente al inicio, cuando se realizó el levantamiento de requisitos para efectuar una exploración preliminar del problema a investigar y justo antes de concebir la interfaz para de esta forma desarrollarla a gusto del cliente.
- **Análisis de documentos:** la revisión de documentos por medio de los cuales se realiza el proceso de encuesta de satisfacción de la calidad de los servicios en la Universidad de Matanzas sede "Camilo Cienfuegos".

De acuerdo con lo planteado anteriormente el documento se estructura de la siguiente forma:

Un primer capítulo referido al **Marco Teórico-Referencial:** se plantean las definiciones fundamentales asociadas al campo de acción. Se hace un estudio sobre las tendencias y tecnologías actuales que serán usadas. Se exponen las características fundamentales de los lenguajes de programación, los sistemas de bases de datos y las características fundamentales de las metodologías de desarrollo de software ágiles.

El capítulo dos para el **Análisis, Diseño de la Solución Propuesta y Construcción:** 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 desarrolla la propuesta para darle solución

a la situación problemática, presentando una planificación por iteraciones.

En el tercer capítulo se aborda la **Validación de la Solución Propuesta**: se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio de los 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.

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.

# 1 Capítulos I Marco Teórico-Referencial

## 1.1 Introducción.

Actualmente en el mundo existen software de gestión que tienen el objetivo de facilitar el trabajo del personal, lograr mayor eficacia en los servicios y ayudar a la investigación de procesos.

Este capítulo contiene los conceptos fundamentales con respecto al dominio del problema, el objeto de estudio, el campo de acción y el análisis de algunas de sus características elementales, con el objetivo de permitir una mejor orientación para su solución. Aborda las tecnologías, herramientas y metodologías que se utilizarán durante el desarrollo de la investigación.

## 1.2 Antecedentes del trabajo

### Características de Lime Survey

Lime Survey es una aplicación web gratuita y de código abierto diseñada para realizar encuestas online o como soporte de encuestas telefónicas. La aplicación resulta adecuada para realizar exámenes online debido a que un examen (especialmente si es de tipo test) no es más que una encuesta en la que se pregunta por conocimientos en lugar de opiniones.

Acerca de la funcionalidad operativa de Lime Survey cabe destacar que:

- Permite crear y administrar distintos niveles de usuarios, con privilegios diferenciados para crear exámenes o consultar resultados.
- Dispone de un nivel de seguridad sofisticado para controlar la contestación. Incluye cookies en el navegador, contraseñas de acceso, activación y desactivación por fecha y hora, filtrado por dirección IP del acceso y otras capacidades.
- Genera contraseñas para los estudiantes y les envía automáticamente correos con las mismas o enlaces personalizados al examen.

- La lista de estudiantes se puede importar desde distintos formatos (base de datos, hoja de cálculo y texto) o directamente validar los accesos desde servidores LDAP.

Las funcionalidades para configurar exámenes, son prácticamente idénticas a las de Moodle, por lo que no vamos a reiterarlas.

En el apartado sobre diseño de las preguntas, cabe destacar las siguientes características:

- El reciclado de preguntas es sencillo. Las preguntas pueden exportarse individualmente y utilizarse en otros exámenes o en ejercicios de auto-evaluación.
- El propio sistema incluye un completo editor similar a un procesador de texto convencional.
- Es posible mostrar ecuaciones complejas escritas en LaTeX (requiere una configuración previa) y se pueden importar las preguntas desde un procesador de texto (Word, OpenOffice) o LaTeX, incluyendo imágenes.

Por último, la funcionalidad de calificación es bastante completa, siendo los aspectos más destacables los siguientes:

- Automatiza la calificación del examen, que puede mostrarse al estudiante nada más terminar el examen.
- La lista de examinandos y sus respuestas pueden exportarse, una vez finalizado el examen, a hojas de cálculo, R o SPSS.

En definitiva, Lime Survey es flexible, fácil de manejar, seguro y cubre adecuadamente las necesidades de un examen real.

### **1.3 Objeto de estudio.**

El objeto de estudio de esta investigación es el proceso de encuestas sobre la satisfacción de la calidad de los servicios, por tanto, se hace necesario hacer una descripción más exacta de este elemento que se pretende transformar y perfeccionar.

### 1.3.1 **Descripción de los conceptos y procesos fundamentales a emplear**

**Las encuestas** son un método de investigación y recopilación de datos, utilizadas para obtener información de personas sobre diversos temas. Las encuestas tienen una variedad de propósitos y se pueden llevar a cabo de muchas maneras dependiendo de la metodología elegida y los objetivos que se deseen alcanzar.

La encuesta es uno de los métodos más utilizados en la investigación de mercado porque permite obtener información real directamente de los consumidores. Por ello, es indispensable que los profesionales de la mercadotecnia e investigadores sepan exactamente la definición de encuesta.

**La entrevista** es un método muy utilizado para recopilar algunos tipos de datos. Una encuesta cara a cara, ya sea a domicilio o en la oficina del encuestado es mucho más costosa que una encuesta telefónica, por ejemplo.

Algunas encuestas combinan varias metodologías. Se puede hacer una encuesta y después hacer una selección de encuestados para una entrevista cara a cara. Los datos de encuestas a veces también se obtienen mediante cuestionarios llenados por los encuestados en grupos, por ejemplo, una clase de escuela o un grupo de compradores en un centro comercial.

#### **Características de una encuesta:**

La necesidad de observar o investigar hechos acerca de una situación nos lleva a realizar una encuesta. Como se mencionó al principio, una encuesta es un método de recopilación de información. ¿Qué se necesita para realizar una encuesta? Primero una muestra conformada por una serie de individuos con características similares, que nos dará referencia sobre lo que queremos saber de una población más grande, de donde precisamente hemos extraído esa muestra. Por ejemplo, una

muestra de votantes se obtiene antes de una elección, para poder conocer lo que opinan los votantes sobre los candidatos.

### **Tipos de encuestas:**

Las encuestas pueden clasificarse de diversas maneras, ya sea por su tamaño y tipo de muestra o por el método utilizado para la recopilación de los datos: vía telefónica, correo, encuestas en papel, encuestas online, entrevistas personales, en fin. Anteriormente eran populares las encuestas por correo postal, pero ahora prácticamente nadie las contesta y envía de regreso, por lo que la tasa de respuesta es demasiado baja. (autores, 2018)

Una **encuesta personal** consiste en una entrevista en la que conversan dos personas, la persona encargada de realizar la encuesta le hace preguntas al encuestado para obtener la información necesaria para la investigación. Dicha encuesta puede ser estructurada (cuestionario formal con preguntas que se les puede dar a todos por igual).

Cuando es una **encuesta en línea** se usa un cuestionario prediseñado con las preguntas que permitan obtener la información necesaria para que en el análisis posterior puedan tomarse las decisiones correctas. Este tipo de encuestas es el ideal para los expertos ya que permite lograr mejores resultados invirtiendo menos tiempo y dinero.

También podemos clasificar a las encuestas por su contenido, pudiendo utilizar preguntas abiertas o cerradas para conocer, por ejemplo, opiniones, actitudes, detalles de un hecho, hábitos, experiencias para una posterior clasificación y análisis de los resultados obtenidos. (autores, 2018)

## **Ventajas de utilizar una encuesta como herramienta de investigación:**

- Permite obtener gran cantidad de información en poco tiempo directamente de los clientes o consumidores.
- Una vez que el cuestionario está hecho no requiere a personal especializado para realizarlo.
- La recolección, procesamiento y análisis de la información es fácil de realizar. (autores, 2018)

### **1.3.2 Flujo actual del proceso involucrado en el campo de acción.**

El proceso de encuesta de satisfacción de la calidad de los servicios en la sede "Camilo Cienfuegos" de la Universidad de Matanzas se realiza de una manera poco fiable, ya que al realizarse en modelos de papel se pierde mucha información. El departamento de calidad de la Universidad elabora las encuestas y las llevan a imprimir en la imprenta, luego estas son revisadas por ellos otra vez y posteriormente son repartidas correspondientemente a cada jefe de departamento, estos últimos se las reparten a los encargados de aplicarlas en sus respectivas áreas y estos las aplican, se las entregan a sus superiores estos a los suyos y así sucesivamente hasta llegar al departamento de calidad de la universidad, donde analizan las respuestas, calculan a través de estas el índice de satisfacción de la calidad y emiten un criterio.

### **1.3.3 Análisis crítico de la ejecución actual de este proceso y las causas que originan la situación problemática, así como sus consecuencias.**

En la sede universitaria "Camilo Cienfuegos" de Matanzas el proceso de encuesta de satisfacción de la calidad de los servicios se lleva a cabo de forma manual, habitual con modelos de papel, no siendo este el método más idóneo para realizar estos procedimientos, puesto que se

dificulta el procesamiento, almacenamiento y control de los datos estadísticos.

Existen demoras en el proceso de encuestas y no se controla la información debido a la forma de guardado y la sistematicidad con se chequean estos aspectos. Todo ello dificulta la obtención de datos estadísticos por lo que se imposibilita dar solución a los aspectos negativos resultantes. Actualmente las encuestas se hacen en hojas de papel, lo que provoca una sobrecarga de trabajo para el responsable de realizar dicha tarea.

## 1.4 **Tendencias tecnológicas.**

### 1.4.1 **Fundamentación de la metodología utilizada.**

Para desarrollar aplicaciones informáticas es preciso seleccionar una metodología que debe ser lo suficientemente adaptable como para poder aplicarse en distintos proyectos, y lo suficientemente sencilla para que no resulte muy engorrosa su utilización, pero a la vez suficientemente completa como para que su utilización por parte de un equipo sea provechosa.

La selección de la metodología a utilizar se hace sobre la base de las características del equipo, las necesidades específicas de la situación y las prioridades demandadas por el cliente. Para elegir una metodología de desarrollo de software se debe tener en cuenta dos factores fundamentales: el tipo de proyecto que se desea desarrollar y el tiempo que se dispone para desarrollar el mismo.

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

**Metodologías ágiles** dan mayor valor a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas. Se basa en la filosofía de que es más importante desarrollar *software* que funcione, que conseguir una buena documentación y es más importante responder ante un cambio, que seguir estrictamente un plan. Se destaca por su popularidad y fácil utilización la metodología *eXtreme Programming(XP)*.

La programación extrema o *eXtreme Programming (XP)* es una metodología para la ingeniería de software formulada por Kent Beck. Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. (Rodriguez, y otros, 2006)

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la Programación Extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. (Ferrer, 2002)

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

Estas características dan motivo para ponerla en práctica en el desarrollo de la aplicación que se desarrolla en este trabajo.

#### 1.4.2 **Arquitectura cliente-servidor**

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes.

Los principales componentes del esquema cliente-servidor son los Clientes, los Servidores y la infraestructura de comunicaciones. En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. (Trygve, y otros, 2009)

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloqueos, la recuperación ante fallas, y otros aspectos afines. (Trygve, y otros, 2009)

Entre las principales características de la arquitectura cliente - servidor, que dan motivo para que se utilice en este proyecto se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz bien definida y única.
- Un servidor puede dar servicios a muchos clientes.
- El cliente proporciona la interfaz entre el usuario y el resto del sistema.

- El cliente necesita solamente conocer la interfaz externa del servidor y no su lógica.
- Los cambios en el servidor implican poco o ningún cambio en el cliente.

#### **1.4.2 Aplicaciones web**

Son consideradas aplicaciones web a aquellas en las que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Con la introducción de Internet y de las Aplicaciones Web, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la web. Afortunadamente, se cuenta con herramientas potentes para realizar esto, ya que han surgido nuevas tecnologías que permiten que el acceso a una base de datos desde la web, por ejemplo, sea un simple trámite. (2002)

Entre las principales ventajas para su empleo en este proyecto se pueden destacar:

- Ahorra tiempo: se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- Existen menos problemas de compatibilidad: basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio significativo en el disco duro.
- Consumo de recursos bajo: dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos del cliente porque se realizan desde otro ordenador.
- Multiplataforma: se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.

- Portables: se accede a través de una página web (sólo es necesario disponer de acceso a la red).
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Disminuye la probabilidad de que los virus dañen los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración: gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.

### 1.4.3 Patrones de diseño

Al conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo de software se les denomina patrones de diseño.

Algunos patrones de diseño utilizados:

**Modelo Vista Controlador (MVC):** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

**Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

**Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

**Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista. (Trygve, y otros, 2009)

Inyección de Dependencias (en inglés *Dependency Injection, DI*) es un

patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. El término fue acuñado por primera vez por Martin Fowler. (Trygve, y otros, 2009)

### 1.4.3 **Lenguajes de programación**

PHP es un lenguaje de programación interpretado, originalmente diseñado para la creación de páginas web dinámicas. Sigue un estilo clásico ya que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. Está más cercano a JavaScript6 o al lenguaje C7, pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, lo que permite acceder a los recursos que este tenga, como es una base de datos y el resultado es enviado al navegador, el cual podría ser una página HTML o de cualquier otro tipo. (2016)

PHP no necesita que el navegador lo soporte, es independiente de este, pero sin embargo para que sus páginas funcionen, el servidor donde están alojadas debe soportar este lenguaje, que tiene numerables ventajas sobre otros lenguajes de programación que se ejecutan de igual manera (como podrían ser los script CGI Perl), como que permite intercalar las sentencias PHP en las páginas HTML. Es el lenguaje ideal para el desarrollo de este proyecto.

HTML: 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). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado.

No tiene nada que ver con un lenguaje de programación, quizás se parezca más al uso de un procesador de texto por la utilización de códigos de comienzo y finalizado de estilo. Además de resultar más sencillo, no se necesita ninguna herramienta de programación, compilador o similar, sino que cualquier editor de texto puede servir para realizar las páginas más sorprendentes. (Raymond, 2003)

Java Scripts: es un lenguaje basado en objetos, muy parecido al Java, pero mucho más sencillo dada su simplicidad sintáctica y su manejabilidad, por lo que casi siempre está dentro de una página HTML. Habitualmente, se utiliza para activar procesos o desarrollar algoritmos que hagan tareas sencillas, (como calculadoras, almanaques, control de errores para formularios...) haciendo la web más viva y dinámica.

Como el HTML no permite activar procesos o desarrollar algoritmos, el Java Script viene muy bien para estas necesidades. No hay que olvidar que está muy limitado frente a lenguajes de programación tradicionales, sobre todo porque no permite la manipulación de ficheros (para evitar problemas de seguridad). Esto ahorra al servidor mucho trabajo pues el código se interpreta en el cliente (desde su navegador), sobre todo se usa para darle vida a las webs haciendo que cambien imágenes, que se procesen datos, entre otros. (Raymond, 2003)

CSS (*Cascading Style Sheets*): las hojas de estilo en cascada son un mecanismo que permite aplicar formato a los documentos escritos en HTML (y en otros lenguajes estructurados, como XML) separando el contenido de las páginas de su apariencia. Para el diseñador, esto significa que la información estará contenida en la página HTML, pero este archivo no debe definir cómo será visualizada esa información. Las indicaciones acerca de la composición visual del documento estarán especificadas en el archivo de la CSS. Lo que posibilita crear páginas web de una manera más exacta. Gracias a

esto el desarrollador es mucho más dueño de los resultados finales de la página.

Framework: en el desarrollo de software es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otras aplicaciones para ayudar a desarrollar y unir los diferentes componentes de un proyecto, es decir, son soluciones completas que llevan incorporado herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

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

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de

gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows. (Zaninotto, y otros, 2007)

Symfony se utilizó porque se ajusta a los siguientes requisitos:

- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos.
- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional:

- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Código fácil de leer que incluye comentarios de *phpDocumentor* y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código ahorrándonos tiempo de trabajo.

JQuery es un framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX, además de ser software libre y de código abierto. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Otra gran ventaja de JQuery es que permite cambiar el contenido de la página web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis. (Leopoldo, 2007)

Bootstrap es un conjunto de herramientas gratuitas para crear sitios web y aplicaciones web. Contiene HTML y basados en CSS plantillas de diseño para tipografía, formas, botones, navegación y otros componentes de la interfaz, así como las extensiones de JavaScript opcionales.

En agosto de 2011, fue lanzado Twitter Bootstrap como código abierto. En febrero de 2012, fue el proyecto de desarrollo de GitHub más popular. Las características que dan motivo para utilizarlo en Bootstrap es compatible con las últimas versiones de los principales navegadores. Algunas de las características que dan motivo a su utilización:

- Desde la versión 2.0 también es compatible con el diseño de respuesta. Esto significa que el diseño de páginas web se ajusta dinámicamente, teniendo en cuenta las características del dispositivo utilizado (de escritorio, tableta, teléfono móvil).
- Bootstrap es de código abierto y está disponible en GitHub. Se anima a los desarrolladores a participar en el proyecto y hacer sus propias contribuciones a la plataforma.
- Recientemente, miembros de la comunidad han traducido la documentación del Bootstrap a varios idiomas, incluido el chino,

español y ruso.

## Ventajas de usar Bootstrap

La más genérica es que permite **simplificar el proceso de maquetación**, sirviéndonos de guía para aplicar las buenas prácticas y los diferentes estándares. Además también:

- Se puede tener una web bien organizada de forma visual rápidamente: la **curva de aprendizaje** hace que su manejo sea asequible y rápido si ya sabes maquetar.
- Permite utilizar muchos elementos web: desde iconos a desplegables, combinando **HTML5, CSS y Javascript**.
- El diseño será **adaptable**, no importa el dispositivo, la escala o resolución.
- El **grid system**: maquetar por columnas nunca fue tan fácil. Además, son muy configurables.
- Se integra muy bien con las principales **librerías Javascript**.
- El haber sido **creado por Twitter** nos da ciertas garantías: está muy pensado y hay mucho trabajo ya hecho. Por lo tanto, hay una **comunidad** muy activa creando, arreglando cosas, ofreciendo plugins y mucho más.
- Cuenta con **implementaciones externas** para WordPress, Drupal, etc.
- Nos permite usar [Less](#), para enriquecer aún más los estilos de la web. (Punto abierto, 2018)

### 1.4.4 Sistema de gestión de bases de datos

Los sistemas de gestión de bases de datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. La función general de un sistema gestor de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se

convertirán en información relevante para una organización.

**MySQL:** es un gestor de base de datos sencillo de usar e increíblemente rápido sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes. Es un sistema cliente-servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios y asegurando el acceso a solo los autorizados. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. Es gratis para aplicaciones no comerciales. Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multi-hilo. Dispone de interfaces de programación de aplicaciones en gran cantidad de lenguajes (C, C++, Java, PHP). Soporta hasta 32 índices por tabla. Implementa una gestión de usuarios y contraseñas que proporciona un buen nivel de seguridad en los datos. Es Software libre (licencia GNU GPL) y Open Source. Trabaja sobre muchas plataformas, incluida Windows, Mac Os X Server, Solaris, Linux, y muchas plataformas de UNIX. Acepta bloqueos y roles de usuario.

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

estos beneficios, que es un servidor que se adecúa perfectamente a las exigencias del cliente.

#### 1.4.5 **Servidor web**

**Servidor Web: Apache HTTP Server** El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Apache tiene amplia aceptación en la red: en el 2005, Apache fue el servidor HTTP más usado, siendo el servidor empleado en el 48% de los sitios Web en el mundo. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. Por lo tanto, las características a destacar para la utilización de este servidor son que:

- El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web.
- Es modular.
- Es open-source: Es un software distribuido y desarrollado libremente.
- Es gratuito.
- Multi-plataforma.

Presenta mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

### **1.5 Conclusiones parciales del capítulo**

Durante este capítulo se profundiza en el análisis de las herramientas existentes hasta el momento, para concluir con los resultados que se esperan alcanzar al final del trabajo. Se define el modelo de desarrollo

a seguir, el cual describe cada artefacto a obtener en cada etapa del desarrollo de esta aplicación y se sientan las bases que dan pie a su inicio.

Queda plasmada de manera clara la necesidad de elaborar una herramienta informática para facilitar el proceso de encuestas sobre la satisfacción de la calidad en la sede universitaria "Camilo Cienfuegos" de Matanzas disminuyendo tiempo y errores en este proceso.

Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología de XP

Se realiza un análisis detallado de las tecnologías a utilizar optando por utilizar el lenguaje de programación web PHP con el framework Symfony y el gestor de bases de datos MySQL.

## 2 Capítulo II Análisis, diseño y construcción de la solución propuesta

### 2.1 Introducción

Para el desarrollo de los sistemas informáticos es preciso tener en cuenta las fases que propone la metodología de desarrollo acorde al tipo de proyecto en el cual se debe lograr cumplir las expectativas del cliente de forma tal que el producto producido tenga la calidad requerida. En este proceso se identifican varias etapas, que van desde la declaración del problema y los requerimientos del sistema, hasta las pruebas y la liberación del mismo. Las metodologías de desarrollo proveen de una guía que ayuda al grupo de desarrollo a organizarse en tiempo, actividades y artefactos a desarrollar (Rivero, 2015)

Entre las metodologías de desarrollo, las ágiles son de más utilidad para el programador, a la cooperación con el cliente y al desarrollo del software con iteraciones cortas, dejando ver su eficacia en proyectos con requisitos cambiantes y cuando se requiere disminuir tajantemente los tiempos de desarrollo manteniendo una alta calidad.

En este capítulo se muestran los elementos precisos para la descripción de la solución deseada. A través de las (HU) Historias de Usuario que albergan las prioridades existentes determinadas por el cliente, es llevado a cabo el análisis de los requerimientos. Por lo anteriormente mencionado se aplica la Metodología XP, pues es una metodología ágil de desarrollo que tiene como características principales (Ferrer, 2002):

**Comunicación:** Los programadores están en constante comunicación con los clientes para cumplir sus requisitos y responder rápidamente en caso de que solicite cambios de los mismos.

- **Simplicidad:** Codificación y diseños simples.
- **Realimentación:** Mediante la realimentación se le otorga al cliente la oportunidad de recibir un sistema que satisfaga sus necesidades.

- **Tenacidad:** Se debe ser tenaz para cumplir los tres puntos anteriores. Hay que tener valor para comunicarse con el cliente y enfatizar algunos puntos a pesar de que esto pueda dar sensación de ignorancia por parte del programador; hay que ser decidido para mantener un diseño simple y no optar por lo que pudiera parecer mejor o un camino más fácil y por último hay que enfatizar que la realimentación será efectiva.

Para determinar si es factible el desarrollo del software se realizará un estudio de factibilidad.

#### 2.1.1 **Etapa de Planificación.**

La planificación se efectúa con el objetivo de alcanzar una eficaz organización del modelo inicial del problema y suministrar así un buen inicio a una solución eficiente. Con este objetivo y según las ideas del cliente sobre el software se desarrollarán las Historias de Usuarios (HU), mediante la cual se conseguirá un origen para el resto de la planificación del proyecto. A la par se ejecutará un estimado de cada una de las entregas del proyecto y del tiempo, basándose en que la planificación inicial se podría ver afectada por cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto.

#### 2.1.2 **Equipo de trabajo y roles.**

Según la metodología XP, se hace necesario ver como se aplican las prácticas para entender más la composición del equipo de trabajo. Estas traducen valores mencionados en el capítulo anterior en actividades que un programador debe realizar diariamente (Rodriguez, y otros, 2006)

Seguidamente se describirá como se aplica al equipo de trabajo formado por dos programadores, el encargado de pruebas y dos gestores:

- Se reafirma el principio de 40 horas de trabajo semanales, con algunos minutos de descanso cada dos o tres horas de trabajo

continuo. Esto contribuye a evitar el cansancio mental de los desarrolladores, y con ello disminuye la probabilidad de errores.

- La retroalimentación con el cliente es un factor muy importante. Los reportes de fallos en el intercambio con este, pueden contribuir a que se detecten errores que pueden pasar por alto a vistas de los desarrolladores.
- El diseño debe ser simple y debe existir un estándar para la notación en el código, esto evita complicaciones en caso de que otra persona desee consultar el código fuente o tal vez los mismos desarrolladores después de algún tiempo.
- Se deben ir programando pequeñas versiones que van aumentando la dimensión poco a poco y en caso de existir fallos se lleva a cabo la reprogramación del código sin variar su funcionalidad.

Tabla 1 Equipo de trabajo y roles

Roles	Nombre
<b>Cliente</b>	Dpto. de calidad de la universidad de Matanzas "Camilo Cienfuegos"
<b>Programador, Tester</b>	Leandro García Benítez
<b>Consultor</b>	Yadrián Pulido García

### 2.1.3 Historias de Usuario iniciales.

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

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

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

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

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.

Las HU son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario.

Estas ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Estas ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se

reúnen para concretar y detallar lo que tiene que hacer dicha historia.  
(González, 2015)

En la tabla 3 se muestra una relación de las HU que se plantearon inicialmente.

Tabla 2 Historias de Usuarios iniciales

No	Nombre	Prior.	Riesgo	Esfuerzo	Iterac.	Entreg.
1	Diseño de la interfaz de usuario	Alta	Alto	3	1	1
2	Diseño y creación de la base de datos	Alta	Medio	2	1	1
3	Gestionar usuarios y roles	Alta	Medio	3	1	1
4	Gestionar encuestas	Alto	Alto	3	2	2
5	Gestionar fórmula	Alto	Medio	3	2	2
6	Generar reportes	Alto	Medio	3	3	3
7	Buscador	Alto	Medio	4	3	3

A continuación, se muestran las HU en detalle para que se pueda comprender el proceso:

Tabla 3 HU 1

HISTORIA DE USUARIO	
Número: <b>1</b>	<b>Usuario:</b> todos
Nombre de Historia: <b>Diseño de la interfaz de usuario</b>	
Prioridad en el negocio: <b>Alta</b>	<b>Riesgo en desarrollo:</b> Alta
Puntos estimados: <b>3</b>	<b>Iteración asignada:</b> 1
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se muestra la vista de la interfaz principal en dependencia del rol del usuario que se ha registrado en el sistema.</b>	

Tabla 4 HU 2

HISTORIA DE USUARIO	
Número: <b>2</b>	<b>Usuario:</b> administrador
Nombre de Historia: <b>Diseño y creación de la base de datos</b>	
Prioridad en el negocio: <b>Alta</b>	<b>Riesgo en desarrollo:</b> Medio
Puntos estimados: <b>2</b>	<b>Iteración asignada:</b> 1
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se diseña e implementa la base de datos en el gestor</b>	

Tabla 5 HU 3

HISTORIA DE USUARIO	
Número: <b>3</b>	<b>Usuario:</b> Administrador
Nombre de Historia: <b>Gestionar Usuarios y Roles</b>	
Prioridad en el negocio: <b>Alta</b>	<b>Riesgo en desarrollo:</b> Medio
Puntos estimados: <b>3</b>	<b>Iteración asignada:</b> 1
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador gestionar los usuarios y asignar los roles correspondientes</b>	

Tabla 6 HU 4

HISTORIA DE USUARIO	
Número: <b>4</b>	<b>Usuario:</b> Administrador
Nombre de Historia: <b>Gestionar encuestas</b>	
Prioridad en el negocio: <b>Alta</b>	<b>Riesgo en desarrollo:</b> Alto
Puntos estimados: <b>3</b>	<b>Iteración asignada:</b> 2
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se le permite al administrador crear, editar, eliminar y publicar las encuestas creadas</b>	

Tabla 7 HU 5

HISTORIA DE USUARIO	
Número: <b>5</b>	<b>Usuario:</b> administrador
Nombre de Historia: <b>Gestionar fórmula</b>	
Prioridad en el negocio: <b>Alta</b>	<b>Riesgo en desarrollo:</b> Alto
Puntos estimados: <b>3</b>	<b>Iteración asignada:</b> 2
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se le permite al administrador crear, editar, eliminar y elegir la fórmula con la que se calculara el resultado final</b>	

Tabla 8 HU 6

HISTORIA DE USUARIO	
Número: <b>6</b>	<b>Usuario:</b> administrador
Nombre de Historia: <b>Gestionar reportes</b>	
Prioridad en el negocio: <b>Alta</b>	<b>Riesgo en desarrollo:</b> Alto
Puntos estimados: <b>3</b>	<b>Iteración asignada:</b> 3
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se le permite al administrador crear un reporte con formato de .pdf con la fecha de publicación y cierre de la encuesta, a quien fue dirigida, preguntas, resultado final y título si esta lo lleva.</b>	

#### 2.1.4 Plan de iteraciones

Este plan es realizado por el desarrollador ya que un buen plan de entregas es directamente proporcional a una correcta planificación en el plan de iteraciones. El plan de Iteraciones se lleva a cabo mediante las historias de usuarios y la estimación del esfuerzo para llevar a cabo cada una de ellas.

En el siguiente gráfico se muestra el Plan de Iteraciones

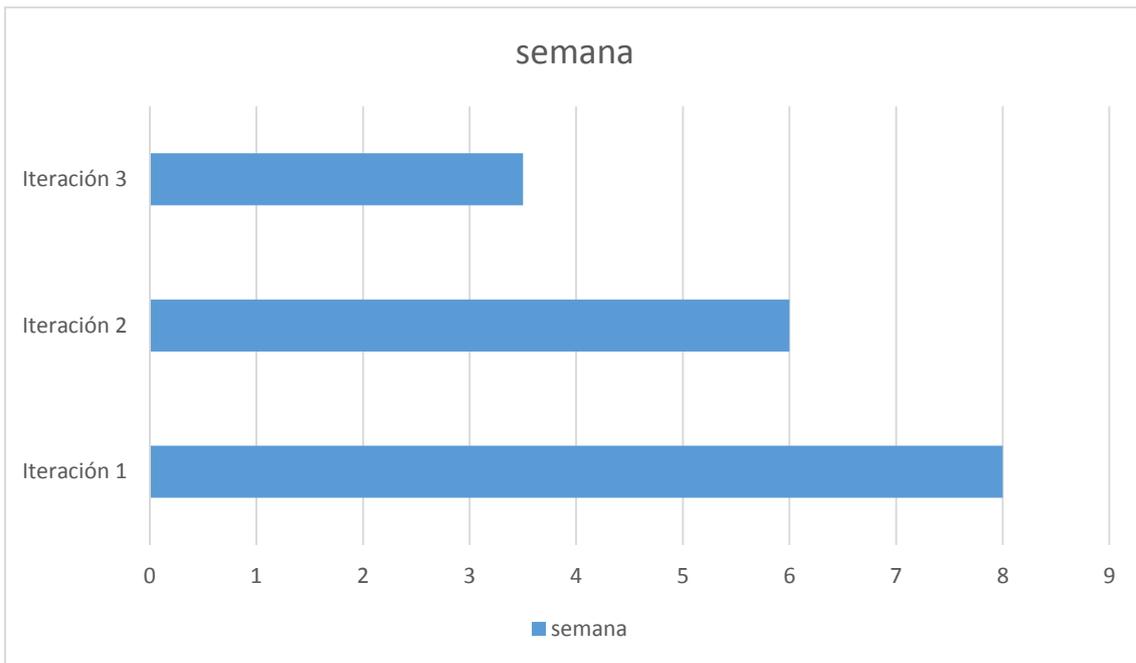


Ilustración 1 Plan de iteraciones

### 2.1.5 Reuniones

La planificación es muy importante en la metodología de desarrollo XP por lo cual es preciso llevar a cabo una revisión continua del plan de trabajo. A pesar de no llevar a cabo una documentación extensa, es muy estricta a la hora de la organización del trabajo. Por lo cual hemos acordado reunirnos semanal con el equipo de trabajo al menos una vez, con el objetivo de seguir el curso de cada iteración.

### 2.1.6 Plan de entregas

Se propone agrupar una o varias iteraciones para que el cliente tenga versiones funcionales del sistema. En la ilustración siguiente se muestra el Plan de Entregas.

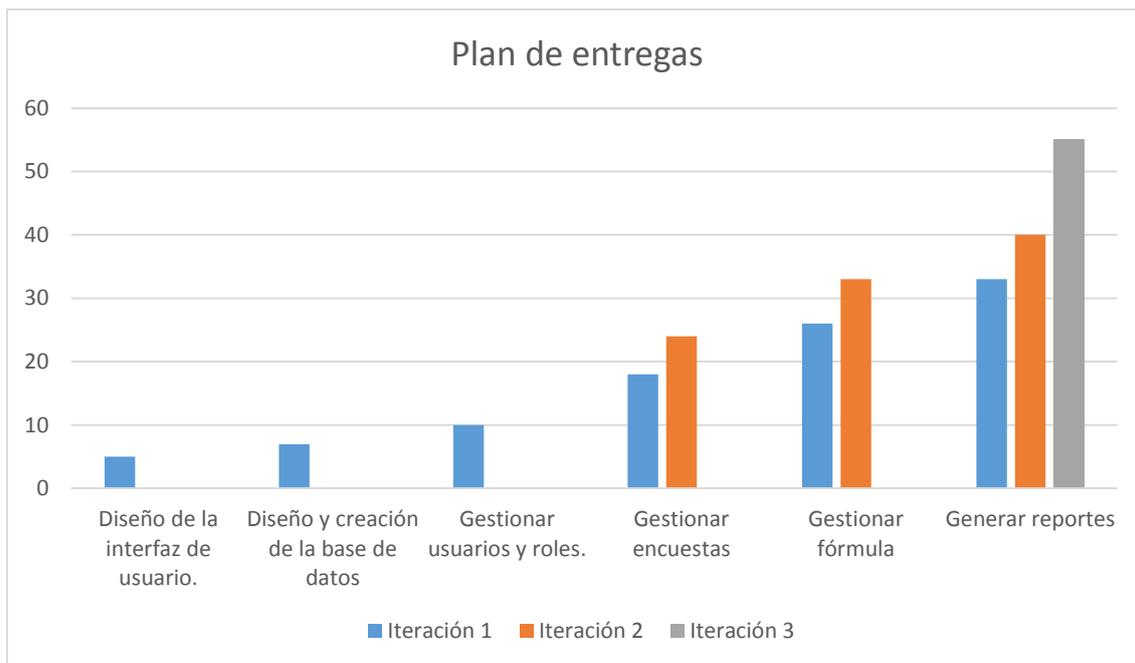


Ilustración 2 Planificación de las Iteraciones en semanas (Elaboración propia)

### 2.1.7 Beneficios tangibles e intangibles.

Este software resultó de gran beneficio para la sede universitaria de Matanzas “Camilo Cienfuegos” puesto que se facilitó todo el proceso de encuesta sobre la satisfacción de la calidad de los servicios, se planteó además una manera estándar para realizarlo y se agilizó este proceso contribuyendo de esta forma que el cumplimiento de los plazos de entrega de los informes se pudiera cumplir.

### 2.1.8 Metáforas

Una metáfora es una abstracción y el diseño de mi sistema corresponde a la aplicación web para el proceso de encuesta sobre la satisfacción de la calidad de los servicios tiene como objetivo brindar una alternativa viable para realizar estas encuestas de una manera más correcta. Además genera informes que son de gran utilidad por el contenido de información que registran acerca de cada operación realizada y te calcula a través de las respuestas de los un índice de satisfacción.

## 2.2 Etapa de diseño.

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos:

- Se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio
- Dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.
- Plantea una nueva visión en el desarrollador de software donde un equipo sencillo en este caso conformado por el tutor y el gestor están en constante comunicación con el cliente y así se obtendrán de mejor forma los resultados esperados.

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

### 2.2.1 Prototipo de interfaz de usuario

A continuación, se muestran una ilustración de la interfaz principal, en el acápite de pruebas se pueden ver otras:

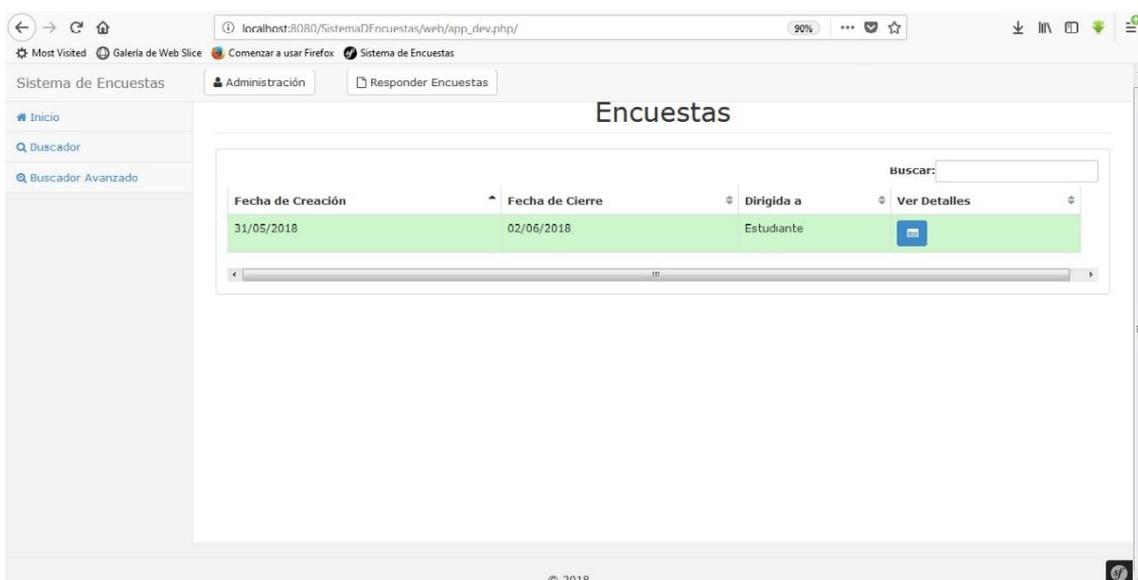


Ilustración 3 Interfaz principal

## 2.2.2 Tareas a desarrollar

Se muestra en la tabla la cantidad de tarea que corresponden a cada HU.

Tabla 9 Tareas de Iteración (Elaboración propia)

No	Nombre HU	No	Tarea de Iteración	Itrc
<b>1</b>	Diseño de la interfaz de usuario.	1	Interfaz Principal	1
		2	Interfaz administración	
		3	Interfaz extranet	
<b>2</b>	Diseño y creación de la base de datos.	4	Diseño de la base de datos.	1
		5	Creación de la base de datos	
<b>3</b>	Gestionar usuarios y roles.	6	Gestionar Usuario	1
		7	Autenticarse	
		8	Gestionar Roles	
<b>4</b>	Gestionar encuestas	9	Crear encuestas	2
		10	Modificar encuestas	
		11	Eliminar encuestas	
		12	Publicar encuestas	
<b>5</b>	Gestionar fórmula	13	Crear fórmula	2
		14	Modificar fórmula	
		15	Eliminar fórmula	
<b>6</b>	Gestionar reportes	16	Crear reporte	3
		17	Modificar reporte	
		18	Generar reporte	
<b>7</b>	Observador	19	Buscador avanzado	3
		20	Buscador sencillo	

A continuación se relacionan algunas tareas de iteración que tenían mayor peso en el desarrollo de este proyecto:

Tabla 10 TI 1

TAREA DE ITERACIÓN	
Número: <b>1</b>	<b>No Historia: 1</b>
Nombre de Tarea: <b>Interfaz principal</b>	
Tipo de Tarea: Desarrollo	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Crear la interfaz visual principal de la aplicación</b>	

Tabla 11 TI 5

TAREA DE ITERACIÓN	
Número: 5	<b>No Historia: 2</b>
Nombre de Tarea: <b>Creación de la Base de Datos</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se creó la Base de Datos empleando a partir de Visual Paradigm con el gestor MySQL</b>	

Tabla 12 TI 7

TAREA DE ITERACIÓN	
Número: 7	<b>No Historia: 3</b>
Nombre de Tarea: <b>Autenticar</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se otorgan los permisos de acuerdo al tipo de rol autenticado en el sistema.</b>	

Tabla 13 TI 9

TAREA DE ITERACIÓN	
Número: 9	<b>No Historia: 4</b>
Nombre de Tarea: <b>Crear Encuestas</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite a los encuestadores crear las encuestas a publicar.</b>	

Tabla 14 TI 10

TAREA DE ITERACIÓN	
Número: 10	<b>No Historia: 4</b>
Nombre de Tarea: <b>Modificar encuesta</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite a los encuestadores modificar las encuestas que se van a publicar en caso de que se les quiera hacer algún cambio.</b>	

Tabla 15 TI 11

TAREA DE ITERACIÓN	
Número: 11	<b>No Historia: 4</b>
Nombre de Tarea: <b>Eliminar Encuestas</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite a los encuestadores eliminar las encuestas que no vayan hacer publicadas.</b>	

Tabla 16 TI 12

TAREA DE ITERACIÓN	
Número: 12	<b>No Historia: 4</b>
Nombre de Tarea: <b>Publicar encuestas</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite a los encuestadores publicar las encuestas.</b>	

Tabla 17 TI 13

TAREA DE ITERACIÓN	
Número: 13	<b>No Historia: 5</b>
Nombre de Tarea: <b>Crear fórmula</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador crear la fórmula adecuada para calcular los resultados de las encuestas</b>	

Tabla 18 TI 14

TAREA DE ITERACIÓN	
Número: 14	<b>No Historia: 5</b>
Nombre de Tarea: <b>Modificar fórmula</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador modificar la fórmula para calcular los resultados de las encuestas</b>	

Tabla 19 TI 15

TAREA DE ITERACIÓN	
Número: 15	<b>No Historia: 5</b>
Nombre de Tarea: <b>Eliminar fórmula</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador eliminar la fórmula que considere innecesaria.</b>	

Tabla 20 TI 16

TAREA DE ITERACIÓN	
Número: 16	<b>No Historia: 6</b>
Nombre de Tarea: <b>Crear reporte</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador crear un reporte con los datos que considere relevante de la encuesta.</b>	

Tabla 21 TI 18

TAREA DE ITERACIÓN	
Número: 18	<b>No Historia: 6</b>
Nombre de Tarea: <b>Generar reporte</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador generar el reporte en formato .pdf</b>	

Tabla 22 TI 19

TAREA DE ITERACIÓN	
Número: 19	<b>No Historia: 7</b>
Nombre de Tarea: <b>Buscador avanzado</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador buscar de una manera avanzada y más específica encuesta publicadas</b>	

Tabla 23 TI 20

TAREA DE ITERACIÓN	
Número: 20	<b>No Historia: 7</b>
Nombre de Tarea: <b>Buscador sencillo</b>	
Tipo de Tarea: <b>Desarrollo</b>	<b>Puntos estimados: 1</b>
Programador responsable: <b>Leandro García Benítez</b>	
Descripción: <b>Se permite al administrador buscar de manera sencilla a través de un periodo de tiempos encuestas publicadas.</b>	

## 2.2.3 Diseño de la base de datos.

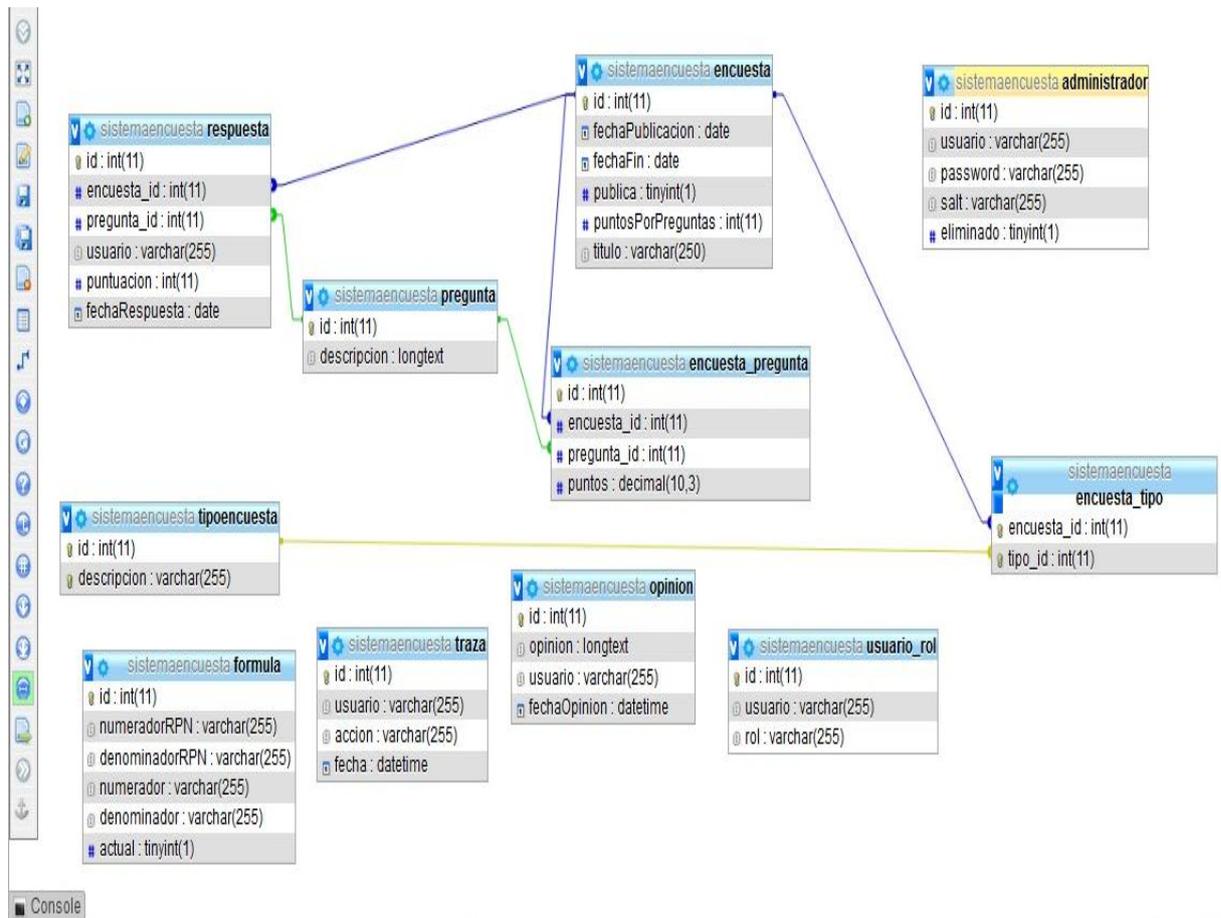


Ilustración 4 Base de datos

### 2.3 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 funcionales. Se escriben las historias de usuario y tareas iniciales que se agrupan en iteraciones y entregas. Al final de las iteraciones se obtuvo una aplicación informática lista para someterse al último proceso de pruebas que se expresa en el siguiente capítulo.

## 3 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

Las pruebas son procesos de ejecución de un programa con la intención de descubrir un error y determinar el status de la calidad de un producto. Los errores pueden empezar a darse desde el primer momento del proceso en que los objetivos pueden estar especificados de forma errónea. Debido a que la comunicación y el trabajo del hombre no son perfectos, el desarrollo del software debe ir acompañado de una actividad que garantice la calidad. 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. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba.

#### 3.2.1 Plan de pruebas

Según 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. (Barrientos, 2014) 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 el plan de pruebas.

*Tabla 24 Plan de pruebas*

No	Nombre de la Historia de Usuario	Pruebas a realizar
1	Diseño y creación de la base de datos	1-Test Base de datos.
2	Diseño y creación de la interfaz de usuario	2-Test Creación de la interfaz de usuario
3	Gestionar usuarios y roles	3-Test autenticarse
4	Gestionar encuestas	4-Test crear encuesta 5-Test eliminar encuesta 6-Test publicar encuesta
5	Gestionar Formula	7-Test crear fórmula 8-Test eliminar formula
6	Generar reporte	9-Test crear reporte
7	Buscador	10-Test buscador sencillo

*Tabla 25: Plan de Pruebas*

### 3.2.2 Pruebas de aceptación

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

En la **tabla** se muestran las clases de equivalencia de las pruebas de aceptación al sistema en la HU.

Tabla 26 pruebas de aceptación

<b>No. 1</b>		<b>Test base de datos</b>
<b>Encargado de Prueba</b>	<b>de</b>	Leandro García Benítez
<b>Descripción</b>		Verifica el funcionamiento de la base de datos
<b>Precondiciones</b>		Estar conectado a la base de datos.
<b>Entradas</b>		Valores para leer o escribir en la base de datos por ejemplo: indicadores mensuales, usuario, contraseña, etc.
<b>Resultados esperados</b>		Se muestran o guardan los datos correctamente
<b>Evaluación</b>		Prueba satisfactoria

Tabla 27 Prueba de aceptación No.2

<b>No. 2</b>		<b>Test creación de la interfaz de usuario</b>
<b>Encargado de Prueba</b>	<b>de</b>	Leandro García Benítez
<b>Descripción</b>		Verificar que se muestren las interfaces visuales implementadas
<b>Precondiciones</b>		Esté corriendo la aplicación
<b>Entradas</b>		Interfaces de la aplicación
<b>Resultados esperados</b>		Se muestren las interfaces visuales de la aplicación
<b>Evaluación</b>		Prueba satisfactoria

Tabla 28 Prueba de aceptación No.3

No. 3		Test autenticarse
<b>Encargado de Prueba</b>	de	Leandro García Benítez
<b>Descripción</b>		Verificar que se autentifique un usuario correctamente en el sistema.
<b>Precondiciones</b>		Estar conectado a la base de datos
<b>Entradas</b>		Nombre de usuario y contraseña
<b>Resultados esperados</b>		Que el usuario se autentifique correctamente
<b>Evaluación</b>		Prueba satisfactoria

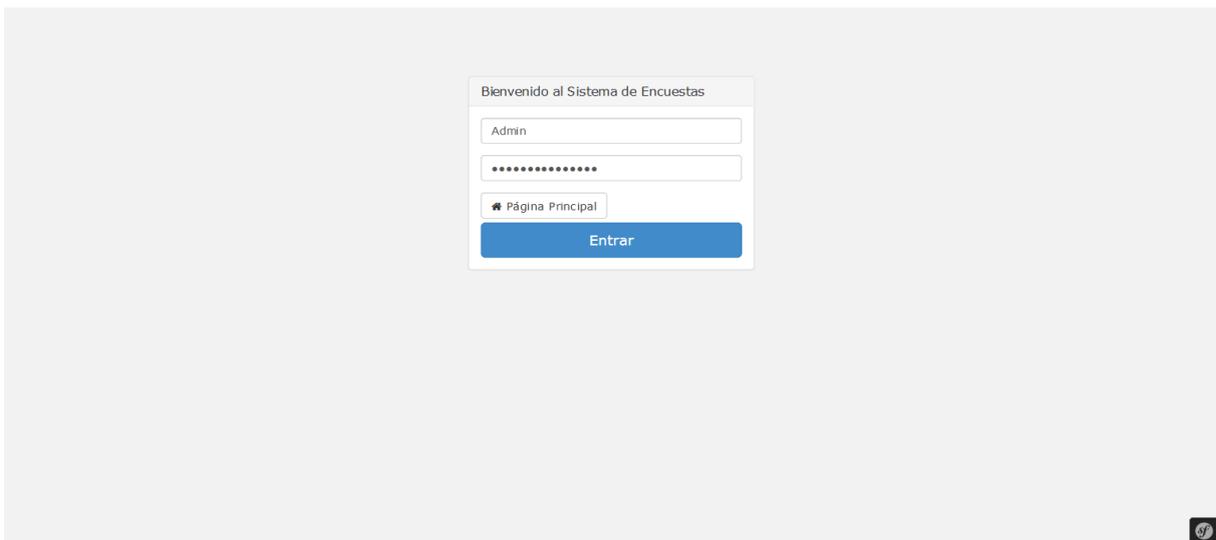


Ilustración 5 Autenticarse

Tabla 29 Prueba de aceptación No.4

No. 4		Test crear encuestas
<b>Encargado de Prueba</b>	de	Leandro García Benítez
<b>Descripción</b>		Verificar que se pueda crear la encuesta
<b>Precondiciones</b>		Estar conectado a la base de datos
<b>Entradas</b>		Elementos de la encuesta
<b>Resultados esperados</b>		Se cree correctamente la encuesta
<b>Evaluación</b>		Prueba satisfactoria

Sistema de Encuestas

- Gestionar Encuesta
- Reporte
- Fórmula
- Ver Opiniones
- Trazas
- Usuarios

**Título:**

**Fecha de Cierre:**  Por favor seleccione una fecha

**Dirigida a:**  
 Estudiante  Profesor  Trabajador No Docente Por favor indique a quien va dirigida la encuesta

**Puntos por Respuestas:**

[Nueva Pregunta](#)

[Usar Pregunta Existente](#)

Por favor inserte al menos una pregunta

Preguntas		
Pregunta	Puntos	Eliminar
<input type="text"/>		<input type="button" value="Eliminar"/>

[+ Crear](#) [Regresar](#)

© 2018

Ilustración 6 crear encuesta

Tabla 30 Prueba de aceptación No.5

No. 5		Test eliminar encuestas
<b>Encargado de Prueba</b>	de	Leandro García Benítez
<b>Descripción</b>		Verificar que se pueda eliminar las encuestas
<b>Precondiciones</b>		Estar conectado a la base de datos y que no esté publicada la encuesta
<b>Entradas</b>		Botón de eliminar
<b>Resultados esperados</b>		Que se elimine correctamente las encuestas
<b>Evaluación</b>		Prueba satisfactoria

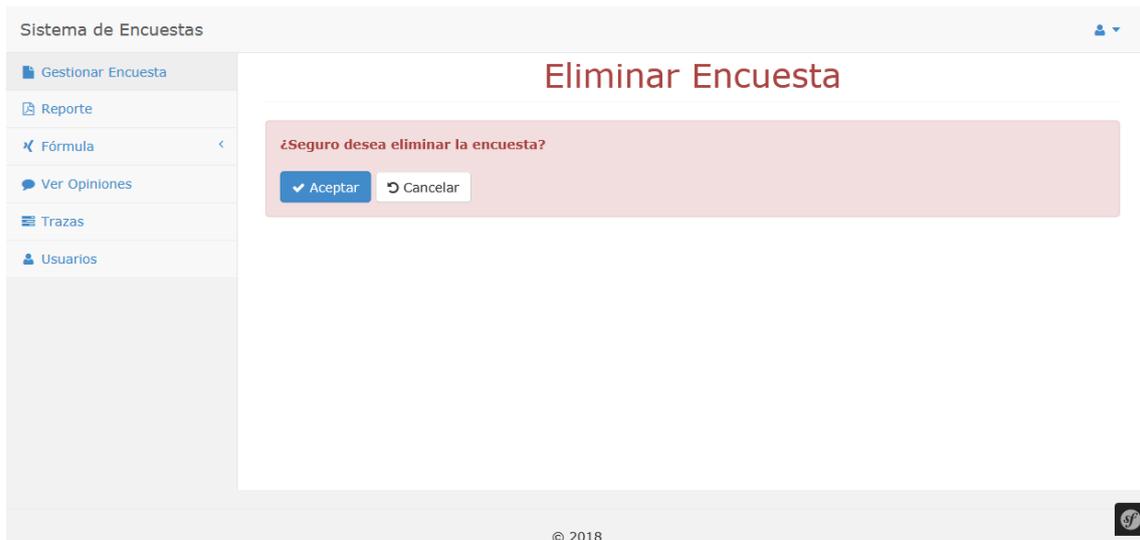


Ilustración 7 eliminar encuesta

Tabla 31 Prueba de aceptación No.6

No. 6		Test publicar encuesta
<b>Encargado de Prueba</b>		Leandro García Benítez
<b>Descripción</b>		Verificar que se pueda publicar la encuesta
<b>Precondiciones</b>		Estar conectado a la base de datos
<b>Entradas</b>		Elementos de la encuestas
<b>Resultados esperados</b>		Que se publique correctamente las encuestas
<b>Evaluación</b>		Prueba satisfactoria



Ilustración 8 publicar encuesta

Tabla 32 Prueba de aceptación No.7

<b>No. 7</b>		<b>Test crear fórmula</b>
<b>Encargado de Prueba</b>	<b>de</b>	Leandro García Benítez
<b>Descripción</b>		Verificar que se pueda crear fórmulas
<b>Precondiciones</b>		Estar conectado a la base de datos
<b>Entradas</b>		Elementos de la fórmula
<b>Resultados esperados</b>		Cree correctamente la formula
<b>Evaluación</b>		Prueba satisfactoria

Tabla 33 Prueba de aceptación No.16

<b>No. 16</b>		<b>Test crear reporte</b>
<b>Encargado de Prueba</b>	<b>de</b>	Leandro García Benítez
<b>Descripción</b>		Verificar que se pueda crear reportes
<b>Precondiciones</b>		Estar conectado a la base de datos
<b>Entradas</b>		Elementos del reporte
<b>Resultados esperados</b>		Se cree correctamente el reporte
<b>Evaluación</b>		Prueba satisfactoria

Tabla 34 Prueba de aceptación No. 17

No. 17		Test buscador avanzado
<b>Encargado de Prueba</b>	<b>de</b>	Leandro García Benítez
<b>Descripción</b>		Verificar que se pueda buscar encuestas
<b>Precondiciones</b>		Estar conectado a la base de datos
<b>Entradas</b>		Datos de la búsqueda
<b>Resultados esperados</b>		Que encuentre correctamente las encuesta
<b>Evaluación</b>		Prueba satisfactoria

### 3.3 Análisis de los resultados.

Después de desarrollar todo el proceso de pruebas con un nivel medio de sencillez se lograron buenos resultados, pues se detectaron diferentes errores, obtenidos principalmente con las realizadas, se solucionaron problemas que evitaban que se cumplieran 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 descripciones acordadas y que además cumple con los requerimientos de rendimiento.

El sistema cumple las expectativas planteadas al inicio del proyecto y complace al cliente, ya que se logra agilizar el proceso de encuesta sobre la satisfacción de la calidad de los servicios y se contribuye sustancialmente al proceso de toma de decisiones.

### 3.4 Beneficios tangibles e intangibles. Estimación del costo.

Este software resultó de gran beneficio para la sede universitaria "Camilo Cienfuegos" de Matanzas, puesto que facilitó el proceso de encuestas sobre la satisfacción de la calidad de los servicios.

La fórmula especificada por Bohem plantea que:

Costo Total (CT) = Tiempo de Desarrollo (TD) \* Salario Mensual (SM)  
\* Cantidad de Hombres (CH)

TD = 25 semanas, lo cual se obtuvo sumando los puntos estimados de las historias de usuarios, agregando el tiempo dedicado a las pruebas. Es aproximadamente 6 meses.

SM = \$500, teniendo en cuenta el salario promedio de un desarrollador en Cuba.

CH = 1, el autor de este trabajo.

CT = \$1000 \* 6 \* 1

CT = \$6000

### **3.5 Conclusiones del Capítulo**

En este capítulo se mostraron los elementos de prueba del sistema, los casos de prueba que se le aplicaron y se explicaron las estrategias de prueba. Las pruebas realizadas utilizando lo anteriormente planteado fueron de gran importancia para demostrar el buen funcionamiento del software y el cumplimiento de los requerimientos del cliente. Como resultado final se pudo obtener una aplicación web, con una apariencia agradable y fácil de usar. La planificación inicial se cumplió a un 100%, se utilizaron las herramientas más actuales para su desarrollo, y el plan de entrega fue cumplido con éxito. El cliente confirma que la aplicación facilita en gran medida el proceso de encuesta sobre la satisfacción de la calidad de los servicios. Por otra parte queda anexado en la documentación del sistema el Manual de Usuario donde se explica claramente cada una de las funcionalidades de la aplicación.

## 4 Conclusiones generales

Como resultado de esta investigación quedaron cumplidos los objetivos trazados arribando a las siguientes conclusiones:

- Los softwares encontrados, vinculados al tema se salen del ámbito de la institución y no le dan solución al problema planteado por lo que no es factible su utilización.
- El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió aportar los elementos necesarios para dar solución a la problemática planteada.
- El uso de la metodología de desarrollo de software XP permitió el diseño, implementación y prueba de una aplicación a la medida de las necesidades del cliente y flexible ante los cambios en los requerimientos.
- El sistema implementado da solución a la situación problemática que lo originó y su explotación significará una mejora considerable en la calidad y eficiencia del procesos que automatiza.

De forma general, se concluye que la aplicación web desarrollada es una herramienta confiable y fácil de utilizar pues eleva la calidad de la gestión, tratamiento y búsqueda de información histórica, reduce los errores a cometer, facilita y perfecciona el proceso. Por estas razones, queda respondida la pregunta científica planteada en la introducción.

## 5 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 web no solo en la provincia, sino desplegarlos al resto de los centros homólogos del país.
- ❖ Propiciar nuevos estudios e investigaciones en aras de lograr un mayor nivel de eficiencia y de informatización de los procesos.
- ❖ Aprovechar las posibilidades de información almacenadas en la aplicación e implementar funcionalidades que respondan a nuevos intereses de la institución.

## 6 Bibliografía

**autores.** ejemplos. [En línea]  
:http://ejemplosde.org/informatica/software-educativo/#ixzz5IE9WQ6mL.

—. **2018.** question pro. [En línea] 2018.  
https://www.questionpro.com/es/encuesta.html.

**Barrientos, Pablo Andrés. 2014.** *Enfoque para pruebas de unidad basado en la generación aleatoria de objetos.* 2014.

**Ferrer, J. 2002.** *Programacion Extrema y Software Libre.* 2002.

**González, Leisy. 2015.** *Sistema Informático de apoyo al control del consumo de electricidad en el MININT, Matanzas.* 2015.

**Krasner, Glenn E. Aug/Sep de 1988.** *A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System.* Aug/Sep de 1988.

**Leopoldo, A. 2007.** 2007.

**2016.** php. [En línea] 2016. php.net.

**2002.** *Programación en Internet: Clientes Web.* s.l. : Editorial Club Universitario., 2002.

punto abierto. [En línea] https://puntoabierto.net/blog/que-es-bootstrap-y-cuales-son-sus-ventajas.

**Punto abierto. 2018.** punto abierto. [En línea] 2018.  
https://puntoabierto.net/blog/que-es-bootstrap-y-cuales-son-sus-ventajas.

**Raymond, Eric. 2003.** *The Art of Unix Programming.* 2003.

**Rivero, Omar. 2015.** *Sistema informático para la automatización del análisis de indicadores económicos por medio de redes neuronales.* 2015.

**Rodriguez, R y Ramírez. 2006.** *Metodologías Ágiles para el desarrollo de Software.* 2006.

**Sánchez, T. y Letelier, et al. 2003.** *Metodologías Ágiles en el desarrollo de Software.* 2003.

**Trygve, Reenskaug y James, Coplien. 2009.** *The DCI Architecture: A New Vision of Object-Oriented Programming.* 2009.

**Zaninotto, F. y F., Potencier. 2007.** *he Definitive Guide to symfony* . 2007.