

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**Desarrollo de un sistema Informático FAQ: Frequently Asked  
Question aplicando al sistema CGWEB de la empresa IT DEL  
ECUADOR**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS**

**MAYRA ELIZABETH VIZCAÍNO RUIZ**

**mayeliza2008@hotmail.com**

**DIRECTOR: ING. LUZ MARINA VINTIMILLA JARAMILLO**

**marina.vintimilla@epn.edu.ec**

**Quito, Febrero 2014**

## **DECLARACIÓN**

Yo, MAYRA ELIZABETH VIZCAINO RUIZ, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración y cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**MAYRA ELIZABETH VIZCAÍNO RUIZ**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por MAYRA ELIZABETH VIZCAINO RUIZ, bajo mi supervisión.

---

Ing. Luz Marina Vintimilla Jaramillo

**DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTOS**

Mi proyecto de titulación ha requerido de gran esfuerzo y dedicación con la supervisión de mi tutora y fundamentalmente del apoyo de las personas que creen en mí y están cerca.

En primer lugar agradezco a Dios por darme la vida, por proveerme de buena salud y agradezco infinitamente a mis padres, mis hermanos y a mi esposo por ser ellos el pilar y el soporte para mi preparación académica, bienestar y por brindarme siempre apoyo incondicional.

Agradezco a mi papá Juan Vizcaíno y a mi madre Eulalia Ruiz a quienes admiro mucho y siempre voy estar eternamente agradecida ya que supieron de la mejor manera inculcarme valores como la dignidad, el respeto, la responsabilidad, amor entre otros.

A mis hermanos Víctor y Johanna por estar siempre conmigo y darme su apoyo incondicional

A mi esposo Gorky Vistin quien me apoyo en los momentos que necesite de su ayuda y compartió parte de mis experiencias y dificultades académicas por las que atravesé

También agradezco a mis amigos los del pasado y el presente por su apoyo incondicional en los momentos de flaqueza e incertidumbre siendo testigos y cómplices a la vez de muchos esfuerzos, alegrías, tristezas y hasta disgustos que compartimos juntos.

A mis maestros de la Escuela Politécnica Nacional agradezco infinitamente por su ayuda, tiempo y dedicación sembrando en mí los conocimientos y formación profesional adquirida.

---

**MAYRA ELIZABETH VIZCAÍNO RUIZ**

## **DEDICATORIA**

Dedico especialmente este trabajo a mis padres y hermanos por brindarme la oportunidad de educarme académicamente e inculcarme valores morales y enseñarme a ser mejor en cada actividad que realice en la vida personal y laboral culminándolas con éxito.

A mi esposo por darme ánimo constante, por ayudarme a sobresalir en los malos momentos que se me presentaron y sobre todo por brindarme amor y comprensión.

A mi tutora por compartir y ser testigo del gran esfuerzo que esta tesis ha requerido, dirigiéndome siempre por el éxito de la misma.

A mis profesores que estuvieron siempre prestos a brindarme su ayuda incondicional para poder desarrollar mi tesis.

---

**MAYRA ELIZABETH VIZCAÍNO RUIZ**

# ÍNDICE DE CONTENIDO

LISTA DE FIGURAS .....	x
LISTA DE TABLAS .....	xi
RESUMEN .....	xii
CAPÍTULO I .....	1
1. AMBIENTACIÓN .....	1
1.1    PLANTEAMIENTO DEL PROBLEMA .....	1
1.2    FOMULACIÓN Y SISTEMATIZACIÓN DEL PROBLEMA .....	2
1.2.1    FORMULACIÓN .....	2
1.2.2    SISTEMATIZACIÓN.....	2
1.3    JUSTIFICACIÓN .....	3
1.4    OBJETIVO GENERAL.....	3
1.5    OBJETIVO ESPECÍFICO.....	3
1.6    ALCANCE .....	4
1.7    PRESUPUESTO.....	4
CAPÍTULO II.....	6
2. MARCO TEÓRICO .....	6
2.1    DEFINICIÓN DE SOFTWARE.....	6
2.2    INGENIERÍA DE SOFTWARE.....	6
2.2.1    PROCESO DEL SOFTWARE .....	7
2.2.2    CARACTERÍSTICAS DEL SOFTWARE .....	7
2.2.2.1 <i>Software se desarrolla o modifica con intelecto, no se fabrica en un sentido clásico</i> .....	7
2.2.2.2 <i>Software no se “desgasta”</i> .....	8
2.2.2.3 <i>Aunque la industria se mueve hacia la construcción de componentes, la mayor parte del software se construye para un uso individualizado</i> .....	8

2.2.3	MANTENIMIENTO DE PROGRAMAS INFORMÁTICOS .....	9
2.3	DOMINIOS DE APLICACIÓN DEL SOFTWARE.....	9
2.3.1	SOFTWARE DE SISTEMAS .....	10
2.3.2	SOFTWARE DE APLICACIÓN.....	10
2.3.3.1	<i>Software basado en la web</i> .....	10
2.4	WEB.....	11
2.4.1	FUNCIONAMIENTO URL DE LA WEB .....	12
2.5	APLICACIÓN DE LA INGENIERÍA DE SOFTWARE EN EL DESARROLLO DEL SISTEMA.....	12
2.5.1	DISEÑO CONCEPTUAL .....	12
2.5.2	DISEÑO LÓGICO .....	13
2.5.3	DISEÑO FÍSICO .....	14
2.6	CAMTASIA ESTUDIO RECOVERY.....	14
2.6.1	PASOS PARA LA GRABACIÓN DE UN VIDEO .....	14
2.7	VISUAL STUDIO 2010.....	15
2.7.1	LO MÁS DESTACADO DE VISUAL STUDIO 2010 .....	15
2.8	VISUAL STUDIO .NET .....	16
2.9	PLATAFORMA .NET.....	17
2.9.1	BIBLIOTECA DE CLASES .NET.....	17
2.9.2	CARACTERÍSTICAS .NET .....	18
2.9.3	.NET FRAMEWORK.....	19
2.10	VISUAL STUDIO 2010 COMO HERRAMIENTAS PARA EL SISTEMA.....	19
2.10.1	INTERFACES DE USUARIOS .....	20
2.10.2	FUNCIONES.....	20
2.11	SQL SERVER 2008 COMO HERRAMIENTA PARA EL SISTEMA.....	20
2.11.1	RELACIÓN ENTRE TABLAS.....	21
2.12	HTML .....	21

2.13	ESTRUCTURAS BÁSICAS DE UN DOCUMENTO HTML .....	22
2.14	HTML 5 .....	23
CAPÍTULO III .....		25
3.	METODOLOGÍA Y MODELO MVC.....	25
3.1	MODELO MVC.....	25
3.1.1	CARACTERÍSTICAS MVC .....	26
3.1.2	MODELO .....	26
3.1.3	VISTA.....	26
3.1.3.1	<i>Distribución</i> .....	27
3.1.3.2	<i>Vistas parciales</i> .....	27
3.1.3.3	<i>Reproductores</i> .....	27
3.1.3.4	<i>Clases de ayuda</i> .....	27
3.1.4	CONTROLADOR .....	27
3.2	MVC Y BASE DE DATOS .....	28
3.3	PARADIGMA DEL MODELO ESPIRAL .....	29
3.3.1	VENTAJAS.....	30
3.3.2	DESVENTAJAS .....	30
3.4	CONSTRUCCIÓN Y ASPECTOS METODOLÓGICOS.....	31
3.5	METODOLOGÍA OMT .....	31
3.5.1	PROCESO DE DESARROLLO OMT.....	32
3.5.1.1	<i>Desarrollo</i> .....	32
3.5.1.2	<i>Conceptualización</i> .....	32
3.5.2	FASES DE LA METODOLOGÍA OMT .....	33
3.5.2.1	<i>Análisis</i> .....	33
3.5.2.2	<i>Diseño del sistema</i> .....	33
3.5.2.3	<i>Diseño de objetos</i> .....	34

3.5.2.4 <i>Implementación</i> .....	34
3.5.2.5 <i>Mantenimiento</i> .....	35
3.5.3 METODOLOGÍA OMT CLASES PARA DESCRIBIR EL SISTEMA.....	35
Modelamiento y diseño de diagramas en base a la metodología OMT con técnicas de modelado UML.....	36
3.5.3.1 <i>Identificación de actores</i> .....	36
3.5.3.2 <i>Diagrama de casos de uso</i> .....	36
3.6.1 MODELO ESTÁTICO .....	36
3.6.1.1 <i>Diagrama de clases</i> .....	37
3.6.1.2. <i>Diagrama de objetos</i> .....	37
3.6.1.3 <i>Diccionario de clases</i> .....	37
3.7.1 MODELO DINÁMICO .....	37
3.7.1.1 <i>Diagrama de interacción</i> .....	38
3.7.1.2 <i>Diagrama de secuencia</i> .....	38
3.7.1.3 <i>Diagrama de colaboración</i> .....	38
3.8.1 MODELO DE PROCESOS .....	38
3.8.1.1 <i>Diagrama de actividades</i> .....	38
3.8.1.2 <i>Diagrama de estados</i> .....	39
3.9 LENGUAJE UNIFICADO DE MODELADO UML .....	39
3.9.1 OBJETIVO UML .....	39
CONCLUSIONES.....	40
RECOMENDACIONES .....	40
BIBLIOGRAFÍA.....	42
RECOMENDACIONES WEB .....	42
LIBROS.....	43

## LISTA DE FIGURAS

FIGURA 1 ( <a href="http://technet.microsoft.com">HTTP://TECHNET.MICROSOFT.COM</a> , 2013) .....	11
FIGURA 2 (MVC, 2012) .....	25
FIGURA 3 (PRESSMAN, 7MA EDICION).....	29

## LISTA DE TABLAS

TABLA 1 (FAQ, 2012).....	5
TABLA 2 (PRESSMAN, UML).....	35

## RESUMEN

Los problemas que se resuelven al desarrollar el sistema Informático FAQ son los de ayudar al usuario en el manejo del sistema de Contabilidad (CGWEB) y cuando se le presente algún problema no será necesario asistir personalmente porque el sistema FAQ podrá encontrar inmediatamente la respuesta

Esta desarrollado mediante el modelo MVC (Modelo Vista Controlador) que es un patrón de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

El proyecto FAQ se encuentra dividido en 3 capítulos además de conclusiones, recomendaciones y bibliografía.

**Capítulo I.-** “Aspectos Generales” planteamiento del problema, objetivos generales y específicos, alcance y el presupuesto para el proyecto.

**Capítulo II.-** “Aspectos Teóricos” se detalla los procesos para desarrollar el sistema mediante la Ingeniería del Software, herramientas de desarrollo utilizadas para la creación del sistema con cada una de sus características.

**Capítulo III.-** “Aspectos Metodológicos” se presentan los conceptos principales del paradigma, la metodología, el modelo y los diagramas aplicados en el desarrollo

**Conclusiones y Recomendaciones.-** Se representan las conclusiones obtenidas al final del proyecto FAQ de igual manera las recomendaciones que se deberán poner en práctica para el buen funcionamiento del sistema.

**Bibliografía.-** Se detalla todo lo referente a libros, referencias web que fueron utilizadas para documentar el desarrollo del sistema.

# **CAPÍTULO I**

## **1. AMBIENTACIÓN**

El capítulo I se encarga de hacer una descripción del sistema financiero CGWEB y la problemática que se les presenta a los usuarios que manejan diariamente el sistema CGWEB de la empresa IT (Información Tecnológica) DEL ECUADOR.

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

El sistema CGWEB es un Sistema Administrativo Financiero que comercializa la empresa IT DEL ECUADOR S.A.

El soporte funcional que brinda la empresa a sus clientes está disponible para los usuarios directamente por la división CG/Web del grupo business T&G y sus colaboradores.

Las llamadas de los usuarios son atendidas durante el horario especificado por cada oficina de soporte. La información actual de contacto puede encontrarse en el sitio web del grupo business T&G.

Ante, una necesidad de soporte funcional el cliente deberá contactar con el Departamento de Soporte correspondiente a su ubicación geográfica, muchas veces las preguntas que realizan son las mismas y deben esperar varios días para poder ser atendidos debido a que deben esperar un turno de atención lo que ha generado una serie de problemas e implicado gastos extras para los usuarios y para la empresa

Los usuarios del sistema CGWEB que necesitan soporte o ayuda de un técnico en sistemas, que les solucione sus problemas rápidamente, evitando quejas constantes de los clientes, por lo tanto se requiere dar soporte funcional y técnico a los usuarios del producto con una herramienta informática que solucione los problemas que se le presentan al usuario. Al implementar el sistema informático FAQ en la página web de la empresa, el usuario podrá encontrar la respuesta que

necesita y además contar con videos como manuales de usuario evitando que el soporte técnico asista personalmente ahorrando tiempo y dinero extra

El sistema FAQ de ayuda al usuario depende de las necesidades del usuario para un buen uso del sistema CGWEB, aumentando la satisfacción de los usuarios de una empresa u organización.

Si la empresa no satisface las necesidades y deseos de sus clientes tendrá una existencia muy corta y un rendimiento muy bajo del sistema provocando que los usuarios no estén satisfechos con el soporte técnico o ayuda inmediata del problema que se les presente al momento de manejar el sistema CGWEB causando a largo plazo pérdida de clientes a la empresa.

## **1.2 FOMULACIÓN Y SISTEMATIZACIÓN DEL PROBLEMA**

### **1.2.1 FORMULACIÓN**

¿Cómo satisfacer las necesidades de los usuarios de la empresa IT del Ecuador en cuanto a las preguntas frecuentes que necesitan ser contestadas lo más pronto posible usando tecnología informática actual?

### **1.2.2 SISTEMATIZACIÓN**

Qué etapas deben ser estructuradas y consideradas para conocer el sistema de funcionamiento para este sistema?

Qué tipo de información y que usuarios son los que se deben considerar?

Cuáles son los requisitos y problemas frecuentes de los usuarios?

Qué tipo de hardware y software son los necesarios además del sistema financiero CGWEB?

### **1.3 JUSTIFICACIÓN**

El desarrollo del sistema FAQ permitirá mejorar el soporte a usuarios, dando respuestas inmediatas a los problemas que se les presente al momento que están manejando el sistema financiero CGWEB.

El Diseño y desarrollo de un sistema Informático FAQ ayuda a llenar un vacío de conocimiento a un determinado problema que se les presenta a los usuarios facilitando el aprendizaje con el uso del sistema CGWEB.

La investigación aporta en la solución de problemas prácticos que se les presenta a las empresas públicas o privadas que han adquirido el sistema CGWEB.

La investigación contribuye a solucionar un determinado problema práctico que se les presenta a los usuarios debido a que la ayuda del sistema informático FAQ tendrá la respuesta a las preguntas más frecuentes, evitando que el soporte técnico asista personalmente

El sistema informático FAQ permite generar diversas aplicaciones prácticas implementándole en la página web de la empresa ahorrando tiempo y dinero.

El interés por aprobar la problemática objeto del diseño y desarrollo del sistema informático FAQ es una preocupación de trabajadores de la empresa IT del Ecuador y se ha visto la necesidad de poder ayudar al usuario inmediatamente evitando visitarle personalmente para solucionar algún problema que se le presente al momento de manejar el sistema CGWEB.

### **1.4 OBJETIVO GENERAL**

Desarrollar un sistema Informático FAQ: frequently asked question aplicando al sistema CGWEB de la empresa IT del Ecuador.

### **1.5 OBJETIVO ESPECÍFICO**

1. Realizar el análisis corporativo de las preguntas frecuentes de los usuarios del sistema CGWEB.

2. Estudiar las especificaciones técnicas y la aplicación del sistema CGWEB.
3. Investigar cual es la asesoría técnica que tiene la empresa.
4. Añadir las respuestas que necesita el usuario mediante videos.
5. Ordenar cronológicamente las preguntas más frecuentes y las respuestas que desea obtener el usuario.
6. Realizar videos debidamente comentados en cada uno de sus pasos.
7. Utilizar herramientas de desarrollo orientada a objetos para la creación del FAQ.
8. Almacenar la información en una base de datos de SQL server 2008.

## **1.6 ALCANCE**

Los problemas que se resuelve al desarrollar el sistema Informático FAQ son los de ayudar al usuario administrador en el manejo del sistema CGWEB y cuando se le presente algún problema no será necesario asistir personalmente porque en el sistema FAQ podrá encontrar inmediatamente la respuesta.

El sistemas FAQ será aplicado en la página web de la empresa IT del Ecuador en el módulo de contabilidad, tendrá restricciones de acceso para el público en general será de uso exclusivo para usuarios registrados y el usuario administrador de dicha empresa.

## **1.7 PRESUPUESTO**

El presupuesto estimado para el proyecto de titulación es de \$4962 dinero que se necesita para las licencias de los programas de desarrollo informático Visual Studio este costo es financiado por la empresa IT DEL ECUADOR S.A.

DESCIPCION		HORAS	VALOR POR HORA	COSTO TOTAL
RECURSO HUMANO	ANALISTA	150	\$6	\$ 900,00
	PROGRAMADOR	300	\$7	\$ 2.100,00
SOFTWARE	SQL 2008			\$ 600,00
	VISUAL STUDIO 2010			\$ 400,00
	OFFICE 2007			\$ 150,00
HARDWARE	COMPUTADOR			\$ 600,00
	IMPRESORA			\$ 80,00
GASTOS VARIOS	FLASH MEMORY			\$ 12,00
	CARTUCHOS IMPRESORA			\$ 10,00
	PAPEL BOND			\$ 10,00
	INTERNET			\$ 100,00
<b>COSTO APROXIMADO DEL SISTEMA</b>				<b>\$ 4.962,00</b>

**Tabla 1 (FAQ, 2012)**

## **CAPÍTULO II**

### **2. MARCO TEÓRICO**

Este capítulo desarrolla el estado de arte de las temáticas que requiere este proyecto como soporte teórico del mismo.

#### **2.1 DEFINICIÓN DE SOFTWARE**

El software de computadora es el producto que construyen los programadores profesionales y que después someten a mantenimiento durante un largo tiempo; incluyen programas que se ejecutan en una computadora de cualquier tamaño o de diferente arquitectura, contenido que se presenta a medida de que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales, cuando se ejecutan llegan a proporcionar las funciones y desempeños buscados (**Pressman R. , Un enfoque practico, 2010**).

Además el software tiene estructuras de datos que permiten que los programas operen en forma adecuada y obtengan la información

#### **2.2 INGENIERÍA DE SOFTWARE**

La ingeniería de software es el establecimiento y el uso de principios fundamentales de la ingeniería con el objeto de desarrollar en forma económica software que sea confiable y trabaje con eficiencia en máquinas reales con una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan para desarrollo de los programas informáticos. Trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos o desarrollos en Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistema de información y en la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software (**Pressman R. , Ing Software, 2010**).

Los métodos de la ingeniería de software son aquellos que nos proporcionan técnicas para la elaboración de software como un producto de calidad que

incluyen un conjunto amplio de tareas como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo.

### **2.2.1 PROCESO DEL SOFTWARE**

Es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo, siendo un enfoque adaptable que permite que las personas que hacen el trabajo busquen o elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán **(Pressman, Proceso del Software, 2010)**.

El proceso de desarrollo del software es aglutinante, une las capas de la tecnología y permite el desarrollo racional y oportuno del software, definiendo una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software.

La estructura establece el fundamento para el proceso de la ingeniería del software por medio de actividades estructurales que son aplicables a todos los proyectos de software sin importar su tamaño o complejidad. Una estructura de proceso general para la ingeniería de procesos consta de cinco actividades comunicación, planeación, modelado, construcción y despliegue, estas se aplican de forma interactiva a medida que avanza el proyecto.

### **2.2.2 CARACTERÍSTICAS DEL SOFTWARE**

El software es un elemento del sistema que es lógico, en lugar de físico. Por tanto el software tiene unas características considerablemente distintas a las del hardware **(Pressman, Características de Software, 2010)**.

#### ***2.2.2.1 Software se desarrolla o modifica con intelecto, no se fabrica en un sentido clásico***

Es decir los proyectos de software no se pueden gestionar como si fueran proyectos de fabricación, la buena calidad se adquiere mediante un buen diseño los costes de realizar el software se encuentran en la ingeniería.

Aunque existen similitudes entre el desarrollo del Software y la construcción del Hardware, ambas actividades son fundamentalmente diferentes porque en el hardware puede introducir problemas de calidad que no existen en el Software.

#### ***2.2.2.2 Software no se “desgasta”***

El software no es susceptible a problemas ambientales que hacen que el hardware se desgaste. Los efectos no detectados harán que falle el programa durante las primeras etapas de su vida, sin embargo una vez corregidas con un buen mantenimiento en los cambios evitara que se produzcan errores. Aunque no se estropea, si puede deteriorarse. Esto sucede debido a los cambios que se efectúan durante su vida.

Por tanto, el mantenimiento del software tiene una complejidad considerablemente mayor que la del mantenimiento del hardware por ejemplo si un componente hardware se estropea, se cambia por otro que actúa como una "pieza de repuesto", mientras que para el software, no es habitual este proceso porque si hay fallas indica un error en el diseño o en el proceso que tradujo el diseño a código ejecutable por la máquina, lo cual significa que el mantenimiento de software incluyen la satisfacción de peticiones de los usuarios.

#### ***2.2.2.3 Aunque la industria se mueve hacia la construcción de componentes, la mayor parte del software se construye para un uso individualizado***

La mayoría del software se construye a medida que evoluciona una disciplina de ingeniería, en vez de ensamblar componentes previamente creados. El componente de software debería diseñarse e implementarse para que pueda volver a ser reutilizado en muchos programas diferentes; los componentes reutilizables modernos encapsulan tanto datos como procesos que se aplican a los datos, permitiendo al desarrollador de software crear nuevas aplicaciones a partir de las partes reutilizables.

La formalización del proceso de desarrollo se define como un marco de referencia denominado ciclo de desarrollo del software o ciclo de vida del desarrollo del software o ciclo de vida del desarrollo. Se puede describir como, "el período de tiempo que comienza con la decisión de desarrollar un producto software y finaliza

cuando se ha entregado éste". Este ciclo, por lo general incluye, una fase de requisitos, fase de diseño, fase de implantación, fase de prueba, y a veces, fase de instalación y aceptación.

### **2.2.3 MANTENIMIENTO DE PROGRAMAS INFORMÁTICOS**

El software no es susceptible a los males del entorno, el software no se estropea pero se deteriora. Durante su vida, el software sufre cambios (mantenimiento), conforme se hacen los cambios, es bastante probable que se introduzcan nuevos defectos, haciendo que el software se vaya deteriorando.

Por tanto el mantenimiento del software tiene una complejidad considerablemente mayor que la del mantenimiento del hardware debido a que es muy importante mejorar el rendimiento del equipo de cómputo, siendo una de las actividades más comunes de la ingeniería de software.

El mantenimiento de software es una actividad muy amplia que incluye la corrección de errores, mejoras de las capacidades, eliminación de funciones obsoletas y optimización. Debido a que el cambio es inevitable, se debe desarrollar mecanismos para la evaluación, controlar y hacer modificaciones.

### **2.3 DOMINIOS DE APLICACIÓN DEL SOFTWARE**

Las categorías de software de computadora plantean retos continuos a los ingenieros de software aplicándose en cualquier situación en la que se haya definido previamente un conjunto específico de pasos, es decir un algoritmo. Algunas veces es difícil establecer categorías genéricas para las aplicaciones del software que sean significativas.

Existen áreas del software que indican la amplitud de las aplicaciones potenciales.

### **2.3.1 SOFTWARE DE SISTEMAS**

El software de sistemas es un conjunto de programas que han sido escritos para dar servicio a otros programas; en un determinado software de sistemas por ejemplo compiladores, editores y herramientas para administrar archivos en el área del software de sistemas se caracteriza por una fuerte interacción con el hardware de la computadora; una gran utilización por múltiples usuarios; una operación concurrente que requiere una planificación, una compartición de recursos y una sofisticada gestión de procesos; unas estructuras de datos complejas y múltiples interfaces externas (**Pressman, Dominios de Aplicación de Software, 2010**).

Denominado también software de base, consiste en programas informáticos que sirven para controlar e interactuar con el sistema operativo, proporcionando control sobre el hardware y dando soporte a otros programas.

### **2.3.2 SOFTWARE DE APLICACIÓN**

Las aplicaciones modernas dentro del área de la ingeniería y las ciencias están abandonando los algoritmos numéricos convencionales

Las aplicaciones que reestructuran son los datos existentes para facilitar las operaciones comerciales o gestionar la toma de decisiones y las tareas convencionales de procesamientos de datos en tiempo real e incluso han tomado características de software de sistemas y además las aplicaciones de software de gestión también realizan cálculo interactivo resolviendo la necesidad específica de negocios.

Es muy importante la recopilación de la mayoría de datos para la creación de un software eficiente.

Llamadas webapps esta categoría de software centrado en redes agrupa una amplia gama de aplicaciones, son un poco más que un conjunto de archivos de hipertexto vinculados que presentan información con uso de textos y gráficas limitadas, con la versión Web 2.0 no solo provee características aisladas, funciones de computo, también provee integración con base de datos corporativas y aplicaciones de negocios

#### ***2.3.3.1 Software basado en la web***

El software basado en web está previsto para asegurar los derechos de acceso dependiendo de las funciones y competencias de cada empleado en una empresa. De esta manera pueden aumentar la eficacia de intercambio de la información

entre los departamentos, dirigir almacenes a través de Internet, realizar reservas y pagos on-line, sistemas de pedidos entre muchos otros.



**Figura 1** (<http://technet.microsoft.com>, 2013)

Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables, la red viene a ser una gran computadora que proporciona un recurso de software casi ilimitado que puede ser accedido por cualquier persona.

## 2.4 WEB

En informática se define como World Wide Web (WWW) o Red informática mundial es un sistema de distribución de información basado en hipertexto o

hipermedios enlazados y accesibles a través de Internet. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces (<http://www.maestrosdelweb.com>, 2013).

#### **2.4.1 FUNCIONAMIENTO URL DE LA WEB**

El primer paso consiste en traducir la parte nombre del servidor en una dirección IP, usando la base de datos distribuida de Internet conocida como DNS. Esta dirección IP es necesaria para contactar con el servidor web y poder enviarle paquetes de datos.

Después se envía una petición HTTP al servidor Web solicitando el recurso. En el caso de una página web típica, primero se solicita el texto HTML y luego es inmediatamente analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros ficheros que formen parte de la página. Las estadísticas de popularidad de un sitio web normalmente están basadas en el número de páginas vistas o las peticiones de servidor asociadas, o peticiones de fichero, que tienen lugar.

Al recibir los ficheros solicitados desde el servidor web, el navegador direcciona a la página tal y como se describe en el código HTML, el CSS y otros lenguajes web. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla.

### **2.5 APLICACIÓN DE LA INGENIERÍA DE SOFTWARE EN EL DESARROLLO DEL SISTEMA**

#### **2.5.1 DISEÑO CONCEPTUAL**

El diseño conceptual se consideró en el análisis de actividades y consistió en la solución de los procesos para el usuario los cuales se expresan con los casos de uso del sistema a desarrollarse.

Cuando se trabaja bajo el análisis conceptual se analiza hechos reales de los cuales se emite un concepto o es posible hacer una idea de ello. Al realizar la

abstracción del FAQ, a nivel informático, es necesario tener los requerimientos formulados por los usuarios con respecto a este. Estos requerimientos contienen el conjunto de hechos y reglas que dan pauta a la creación del esquema conceptual donde por medio de este se podrá realizar una descripción de alto nivel de la futura base de datos (<http://www.unalmed.edu.com>, 2013).

Para manipular este esquema se utiliza un modelo conceptual que proporciona un lenguaje que permite utilizar un conjunto de símbolos (estándares) para la creación de este.

El diseño conceptual se hace independiente al sistema gestor de base de datos (DBMS) que utilice el usuario para la implementación de esta.

Este diseño consistió de las siguientes tareas:

- Identificar a los usuarios.
- Validar usuarios que pueden acceder al sistema CGWEB.
- Obtener preguntas y respuestas más frecuentes de los usuarios.
- Estudiar la información recopilada con el personal de la empresa que capacita a los usuarios.
- Documentar Preguntas y Respuesta.
- Para la obtención de estos requerimientos se realizaron encuestas y visitas a los usuarios obteniendo así una solución adecuada para el desarrollo del sistema FAQ.

## **2.5.2 DISEÑO LÓGICO**

El diseño lógico una vez establecido el modelo conceptual del problema o situación, permite que estos se puedan representar usando de manera eficiente posibles recursos para estructurar datos y modelar restricciones disponibles en el modelo lógico. El diseño lógico refina, organiza y detalla la solución del proceso además define formalmente las reglas y políticas específicas del proyecto (**Pressman, Modelo de procesos, 2010**).

Este diseño fue utilizado para realizar las siguientes actividades:

- Identificar y definir los objetos del negocio y sus servicios.
- Definir las interfaces.

- Identificar las dependencias entre objetos.
- Revisar preguntas y respuestas cuantas veces sea necesario.

### **2.5.3 DISEÑO FÍSICO**

El diseño físico traduce al diseño lógico a una solución de implementación y costo efectiva o económica conveniente (Pressman, *Modelo de procesos*, 2010).

Las tareas utilizadas para el desarrollo del sistema dentro de este diseño fueron:

- Definir los componentes.
- Refinar el comportamiento y distribución de componentes.
- Especificar las interfaces de los componentes.
- Distribuir los repositorios físicos de datos.
- Examinar la tolerancia a fallas y la recuperación de errores.
- Validar el diseño físico.

## **2.6 CAMTASIA ESTUDIO RECOVERY**

Camtasia Estudio es una aplicación de software para la creación de tutoriales en vídeo y presentaciones, el usuario puede usar Camtasia para grabar la actividad en pantalla, personalizar y editar el contenido agregando elementos interactivos, y compartir con diferentes personas. Producir videos interactivos es como hacer clic en enlaces, tablas de contenido, búsquedas y más.

### **2.6.1 PASOS PARA LA GRABACIÓN DE UN VIDEO**

El presentador lleva a cabo todos los pasos de la demostración en secuencia y sin interrumpir el proceso de grabación puede saltar de una aplicación a otra. Además es capaz de detener la grabación con una tecla de acceso directo combinación en cualquier momento y aplica las configuraciones definidas, tales como cuándo y cómo mostrar el cursor. Las nuevas versiones de Camtasia ofrece la opción de dibujar un círculo alrededor del cursor cuando se hace clic en el ratón, para ilustrar mejor las acciones del presentador.

Una vez que la presentación ha sido grabada, el usuario es capaz de revisar por el corte y / o pegar las diferentes partes, según sea necesario. Además, Camtasia Studio permite superposición de voz para el presentador, así como la posibilidad de añadir efectos de sonido y música. Camtasia permite la grabación de audio, mientras que la pantalla de captura está en marcha.

El programa Camtasia Studio permite varios tipos de archivos de vídeo por ejemplo, mp4, WMV (Windows Media Video), etc que se almacenan en su propio formato Camtasia propietario, que sólo es legible por Camtasia, este formato permite tamaños de archivo más pequeños, incluso para los videos más largos. Camtasia también permite que el vídeo creado para ser exportado a formatos de vídeo comunes.

## **2.7 VISUAL STUDIO 2010**

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET

Visual Studio 2010 ofrece herramientas de desarrollo avanzadas, funciones de debugging, funciones para bases de datos y funciones innovadoras que permiten crear rápidamente aplicaciones futuras para distintas plataformas. Visual Studio 2010 incluye mejoras como diseñadores visuales para un desarrollo más rápido con .NET Framework 4 mejoras sustanciales en las herramientas de desarrollo Web y mejoras de programación que aceleran el desarrollo a partir de todo tipo de datos.

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. Aunque actualmente se han desarrollado las extensiones necesarias para muchos otros Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma.

### **2.7.1 LO MÁS DESTACADO DE VISUAL STUDIO 2010**

**Mejoras visuales** Se ha rediseñado el IDE con el fin de mejorar la legibilidad. Se han quitado las líneas y los degradados innecesarios para conseguir una mayor claridad.

**Compatibilidad con distintos monitores** Las ventanas de documento, como la ventana Editor de código y Vista de diseño, pueden situarse ahora fuera de la ventana del IDE. Por ejemplo, puede arrastrar el editor de código fuera del IDE para poder verlo en paralelo a la ventana Vista de diseño.

### **Herramientas para ayudar a explorar el código con rapidez**

**Editor de código** El nuevo editor de código facilita la lectura del código. Puede acercarse al texto si presiona CTRL y mueve la rueda del mouse. Además, al hacer clic en un símbolo en Visual C# o Visual Basic, todas las instancias de ese símbolo se resaltan automáticamente.

**Búsqueda mientras se escribe** La nueva característica Navegar a ofrecer compatibilidad con la búsqueda mientras se escribe para archivos, tipos y miembros. Navegar a le permite aprovechar el uso combinado de mayúsculas y minúsculas y los subrayados para abreviar su texto de búsqueda.

**Jerarquía de llamadas** En Visual C# y Visual C+++, Jerarquía de llamadas le permite navegar desde un miembro hasta los miembros que lo llaman y hasta los miembros a los que llama. Esto resulta útil para explorar código orientado a objetos.

**Depuración** Una ventana Subprocesos rediseñada proporciona filtrado, búsqueda y expansión de pilas de llamadas y agrupación. Además, ahora puede organizar y buscar puntos de interrupción y compartirlos con otros desarrolladores. Para obtener más información. Con las nuevas ventanas Pilas paralelas y Tareas paralelas, resulta más fácil ver y depurar el código paralelo escrito en C++, C# o Visual Basic

## **2.8 VISUAL STUDIO .NET**

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. Actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web y dispositivos móviles (<http://www.visualstudio.com/>, 2013).

## **2.9 PLATAFORMA .NET**

El objetivo de la plataforma .NET es simplificar el desarrollo de aplicaciones Web.

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

.NET ofrece una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

Esta plataforma además soporta los estándares sobre los cuales se basan los servicios Web.

### **2.9.1 BIBLIOTECA DE CLASES .NET**

La Biblioteca de Clases Base (BCL por sus siglas en inglés) maneja la mayoría de las operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones, incluyendo entre otras:

- Generación de código
- Auto descripción de código
- Manejo de datos (ADO.NET)
- Administración de memoria
- Transmisión y recepción de datos por distintos medios (XML, TCP/IP)
- Administración de componentes Web que corren tanto en el servidor como en el cliente (ASP.NET)
- Manejo del sistema de ventanas

- Herramientas de seguridad e integración con la seguridad del sistema operativo
- Interacción con otras aplicaciones
- Manejo de arreglos de datos y colecciones
- Manipulación de archivos de imágenes
- Compilación de código, etc.

## 2.9.2 CARACTERÍSTICAS .NET

Es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

**Cargador de clases:** permite cargar en memoria las clases.

**Compilador MSIL a nativo:** transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.

**Administrador de código:** coordina toda la operación de los distintos subsistemas del Common Language Runtime.

**Recolector de basura:** elimina de memoria objetos no utilizados.

**Motor de seguridad:** administra la seguridad del código que se ejecuta.

**Motor de depuración:** permite hacer un seguimiento de la ejecución del código aun cuando se utilicen lenguajes distintos.

**Verificador de tipos:** controla que las variables de la aplicación usen el área de memoria que tienen asignado.

**Administrador de excepciones:** maneja los errores que se producen durante la ejecución del código.

**Soporte de multiproceso (hilos):** permite ejecutar código en forma paralela.

**Empaquetador de COM:** coordina la comunicación con los componentes

COM para que puedan ser usados por el .NET Framework.

**Soporte de la Biblioteca de Clases Base:** interfaz con las clases base del

.NET Framework. Esto quiere decir que existen tipos de estructuras como es la de java y la .NET

### **2.9.3 .NET FRAMEWORK**

Es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este framework siendo fácil de usar por los desarrolladores.

Tiene una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones; mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables (<http://www.slideshare.net>, 2013)

.Net Framework soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C# (C Sharp), Visual Basic o C++ hasta otros lenguajes menos conocidos como Perl o Cobol (<http://www.microsoft.com>, 2013).

## **2.10 VISUAL STUDIO 2010 COMO HERRAMIENTAS PARA EL SISTEMA**

Visual Studio 2010 es la herramienta, acompañada por .NET Framework 4.0. La fecha del lanzamiento fue el 12 de abril de 2010. Es un lenguaje de programación visual que realiza un gran número de tareas sin la necesidad de escribir código, ya que tiene un fácil manejo de operaciones graficas que se las puede realizar con la utilización del Mouse.

Visual 2010 es un programa basado en objetos, es decir es un lenguaje de programación orientado a objetos facilitando los mecanismos de herencia y polimorfismo; gracias a esto el desarrollo del sistema es más fácil de comprender

y de implementa. .Dentro del desarrollo del sistema el visual 2010 fue utilizado en los siguientes términos:

### **2.10.1 INTERFACES DE USUARIOS**

La interfaz de usuario es el vínculo entre el usuario y el programa de computadora. Una interfaz es un conjunto de comandos o menús a través de los cuales el usuario se comunica con el programa.

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora. La mejor interacción humano-máquina a través de una adecuada interfaz (Interfaz de Usuario), que le brinde tanto comodidad, como eficiencia.

Para la creación de las interfaces del sistema se tomó en cuenta las necesidades o requerimientos de los usuarios de tal forma que satisfaga las necesidades de los mismos creando una interfaz fácil de aprender y usar (<http://www.fismat.umich.mx>, 2013).

### **2.10.2 FUNCIONES**

Es un conjunto de instrucciones que permiten procesar las variables para obtener un resultado, las funciones nos permiten su utilización sistemática tantas veces como queramos sin necesidad de escribir las instrucciones tantas veces como veces se desee utilizarla.

Las funciones fueron de gran utilidad para el desarrollo del sistema ya que la mayor parte del sistema consiste en una secuencia repetitiva de operaciones o procesos.

## **2.11 SQL SERVER 2008 COMO HERRAMIENTA PARA EL SISTEMA**

SQL Server 2008 es capaz de gestionar cualquier tipo de datos, en cualquier sitio y momento. Almacena datos de documentos estructurados, semi estructurados o

no estructurados como imágenes, música y archivos directamente dentro de la base de datos.

Se obtiene más rendimiento de los datos, poniendo a disposición servicios integrados como son consultas, búsquedas, sincronizaciones, informes y análisis.

Se utilizó SQL SERVER 2008 por ser una plataforma actualizada y difundida para el manejo de datos de una manera segura, simple y muy potente. La utilización que se dio dentro del sistema fue correr, modificar, insertar y borrar registros de las tablas de datos; además de contar con las seguridades adecuadas para el manejo de la información de las preguntas más frecuentes que formulaban los usuarios y técnicos de soporte de la empresa.

### **2.11.1 RELACIÓN ENTRE TABLAS**

Las relaciones entre tablas representan las entidades que se consideran interesantes en la base de datos del sistema FAQ, cada instancia de la entidad encontrará sitio en una línea de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad.

Las líneas en una relación son una colección no ordenada de elementos diferentes. Para distinguir una línea de otra, se recurre al concepto de "llave primaria", o sea a un conjunto de atributos que permiten identificar unívocamente en una relación.

En una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una línea, pero entre éstas se elegirá una sola para utilizar como llave primaria. Los atributos de la llave primaria no pueden asumir el valor nulo (que significa un valor no determinado), en tanto que ya no permitirían identificar una línea concreta en una relación. Esta propiedad de las relaciones y de sus llaves primarias está bajo el nombre de integridad de las entidades.

Los motivos de utilizar relaciones entre tablas en la base de datos del sistema son:

- Ofrecer un sistema simple y eficaz para representar y manipular los datos.
- Se analizó en un modelo, el relacional, con sólidas bases teóricas y practicas

## **2.12 HTML**

Siglas de HyperText Markup Language hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Este lenguaje no describe la apariencia de un documento sino que ofrece a cada plataforma la información para que de formato según su capacidad y la de su navegador (tamaño de la pantalla, fuentes que tiene instaladas, etc.). Por eso es importante, diseñar los documentos con un contenido claro y bien estructurado que resulte fácil de leer y entender en cualquier navegador (<http://www.ri5.com.ar/ayuda03.php>, 2013).

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 (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado.

## 2.13 ESTRUCTURAS BÁSICAS DE UN DOCUMENTO HTML

Las "tags" que describen la estructura general de un documento y dan una información sencilla sobre él son: <HTML> <HEAD> <TITLE> <BODY>. Estas tags no afectan a la apariencia del documento y solo interpretan y filtran los archivos HTML

Un documento HTML comienza con la etiqueta <html>, y termina con </html>. Dentro del documento (entre las etiquetas de principio y fin de html), hay dos zonas bien diferenciadas: el encabezamiento, delimitado por <head> y </head>, que sirve para definir diversos valores válidos en todo el documento; y el cuerpo, delimitado por <body> y </body>, donde reside la información del documento.

**<HTML>**: Limitan el documento e indica que se encuentra escrito en este lenguaje.

**<HEAD>**: Especifica el prólogo del resto del archivo. Son pocas las "tags" que van dentro de ella, destacando la del título

**<TITLE>** que será utilizado por los marcadores del navegador e identificará el contenido de la página. Solo puede haber un título por documento, preferiblemente corto aunque significativo, y no caben otras tags dentro de él. En head no hay que colocar nada del texto del documento.

La única utilidad del encabezamiento en la que nos detendremos es la directiva <title>, que permite especificar el título de un documento HTML.

```
<title>FAQ</title>
```

**<BODY>**: Encierra el cuerpo principal del documento, el contenido.

El cuerpo de un documento HTML contiene el texto que, con la presentación y los efectos que se decidan, se presentará ante el hiperlector. Dentro del cuerpo son aplicables todos los efectos que se van a mencionar en el resto de esta guía. Dichos efectos se especifican exclusivamente a través de directivas. Esto quiere decir que los espacios, tabulaciones y retornos de carro que se introduzcan en el fichero fuente no tienen ningún efecto a la hora de la presentación final del documento.

La estructura básica de un documento HTML queda de la forma siguiente:

```
<html>
```

```
<head>
```

```
<title>Título</title>
```

```
</head>
```

```
<body>
```

Texto del documento, menciones a gráficos, enlaces, etc.

```
</body>
```

```
</html>
```

## 2.14 HTML 5

Es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml) Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo (<http://www.w3schools.com>, 2013)

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a

las etiquetas `<div>` y `<span>`, pero tienen un significado semántico, como por ejemplo `<nav>` (bloque de navegación del sitio web) y `<footer>`. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos `<audio>` y `<video>`.

Mejoras en el elemento `<canvas>`, capaz de renderizar en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE) elementos 3D.

Algunos elementos de HTML 4.01 han quedado obsoletos, incluyendo elementos puramente de presentación, como `<font>` y `<center>`, cuyos efectos son manejados por el CSS. También hay un renovado énfasis en la importancia del scripting DOM para el comportamiento de la web. 2.0.

## CAPÍTULO III

### 3. METODOLOGÍA Y MODELO MVC

#### 3.1 MODELO MVC

El patrón de diseño MVC “Model View Controller” o “Modelo Vista Controlador” provee un mecanismo que posibilita separar los datos (el modelo) de la forma en que estos serán visualizados (la vista).

Algunos de los componentes que permiten visualizar información implementan este patrón de diseño para no obligar al usuario a que los datos que quiere mostrar estén almacenados sobre un determinado tipo de soporte.

Explicado de otro modo un componente por ejemplo lista puede mostrar objetos almacenados en un tipo de datos que asocia valores con claves (array), en un vector, en un archivo, o en una tabla de una base de datos. El componente que es la vista está totalmente independizado del modelo o soporte de los datos (Sznajdleder, 2010).



Figura 2 (MVC, 2012)

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

### **3.1.1 CARACTERÍSTICAS MVC**

El patrón MVC fue una de las primeras ideas en el campo de las interfaces gráficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones

MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posteriormente el mantenimiento del sistema (<http://www.yiiframework.com>, 2013).

### **3.1.2 MODELO**

Los modelos son a menudo compartidos entre los diferentes sub-aplicaciones de una aplicación web teniendo una estructura de datos subyacente que contiene propiedades para representar datos específicos;

El modelo debe contener la lógica de negocio (por ejemplo, reglas de validación) para asegurar que los datos representados cumple con el requisito de diseño; puede contener código para la manipulación de datos. En general, los modelos no deben contener lógica que trata directamente con los usuarios finales.

Los modelos no deben usar GET, POST, u otras variables similares que están directamente vinculados a la solicitud del usuario final.

### **3.1.3 VISTA**

Representan los modelos en el formato que los usuarios finales desean. En general, considera que debe contener todo el código de presentación, tales como HTML o código PHP sencillo de recorrer, el formato y los datos creados deben evitar contener código que realiza consultas explícitas en la BDD es decir deben evitar el acceso directo a GET, POST, y otras variables similares que representan a la solicitud del usuario final.

La vista se debe centrar en la pantalla y el diseño de los datos facilitados a la misma por el controlador y / o modelo, pero no intentar acceder a una demanda de las variables o la base de datos directamente, puede acceder a las propiedades y métodos de los controladores y modelos directamente. Sin embargo, esto debe hacerse sólo con el propósito de presentación (Sznajdleder, Estudio del lenguaje y desarrollo de aplicaciones, 2010).

Las vistas pueden ser reutilizadas en diferentes formas:

#### ***3.1.3.1 Distribución***

Son áreas comunes de presentación dependiendo del diseño como la página de encabezado, pie de página, imágenes del sistema

#### ***3.1.3.2 Vistas parciales***

Se utiliza las vistas parciales para puntos de vista que no están decoradas con diseños es decir para reutilizar fragmentos de código de presentación. Por ejemplo, utilizamos vista parcial form.html para hacer que el modelo de formulario de entrada que se utiliza tanto en la creación de modelos y actualización de páginas.

#### ***3.1.3.3 Reproductores***

Se necesita una gran cantidad de lógica para presentar una visión parcial, la visión parcial se puede convertir en un reproductor cuya clase de archivo es el mejor lugar para contener esta lógica. Para aparatos que generan una gran cantidad de código HTML, lo mejor es usar ver archivos específicos para el reproductor.

#### ***3.1.3.4 Clases de ayuda***

En vistas a menudo necesitamos algunos fragmentos de código para realizar tareas pequeñas como formato de datos o generar etiquetas HTML. En lugar de colocar el código directamente en los archivos de vista, un mejor enfoque consiste en colocar todos estos fragmentos de código en una clase de ayudante de vista, después tiene que utilizar la clase de ayuda en sus archivos de vista.

Clases de ayuda se pueden poner en un directorio auto cargable para que puedan ser utilizados sin la inclusión de una clase explícita.

### **3.1.4 CONTROLADOR**

Responde a eventos es decir a acciones del usuario) e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo en el

sistema FAQ puede editar un documento o un registro de alguna pregunta en la base de datos, además puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta de modelo como por ejemplo, desplazamiento o por los diferentes registros de una base de datos), por tanto se podría decir que el controlador hace de intermediario entre la 'vista' y el 'modelo'.

Por lo tanto, controladores puede acceder a GET, POST y otras variables de HTML que representan solicitudes de los usuarios; puede crear instancias de modelo y gestionar su ciclo de vida.

En una aplicación MVC bien diseñado, los controladores son a menudo muy finos, que contiene probablemente sólo unas pocas docenas de líneas de código, mientras que los modelos son muy grandes, que contiene la mayor parte del código responsable de la representación y manipulación de los datos. Esto se debe a la estructura de datos y lógica de negocio representado por modelos es típicamente muy específica para la aplicación particular, y debe ser muy personalizada para satisfacer los requisitos específicos de la aplicación, mientras que la lógica del controlador a menudo sigue un patrón similar en todas las aplicaciones y por lo tanto puede así ser simplificada por el marco subyacente o las clases base (<http://www.yiiframework.com>, 2013)

## **3.2 MVC Y BASE DE DATOS**

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre la Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

### 3.3 PARADIGMA DEL MODELO ESPIRAL

Del modelo espiral o un proceso evolutivo con una separación no rígida entre las fases del desarrollo.

OMT se puede adaptar para utilizar en diversos órdenes de sus actividades importantes para permitir que la ejecución simultánea en el trabajo sea realizada y la interacción de actividades y de sus tareas. Así ajustes de OMT agradable en un marco modelo espiral de ciclo vital del desarrollo incremental del software lógica.



**Figura 3 (Pressman, 7ma Edición)**

Cuando empieza el proceso evolutivo, el equipo de ingeniería del software gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. El primer circuito de la espiral produce el desarrollo de una especificación de productos; los pasos siguientes en el modelo espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software. Cada paso de la región de planificación produce ajustes en el plan del proyecto. El coste y la planificación se ajustan según la reacción ante la evolución del cliente.

El modelo espiral se divide en un número de actividades de marco de trabajo, también llamadas regiones de tarea, cada una de las regiones están compuestas por un conjunto de tareas del trabajo llamado conjunto de tareas que se adaptan a

las características del proyecto que va a emprenderse, en todos los casos se aplican actividades de protección.

### **3.3.1 VENTAJAS**

- El modelado espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora, no terminal cuando se entrega el software.
- Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- Permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.
- Demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto.
- Reduce los riesgos antes de que se conviertan en problemáticos.
- El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos.
- Incorpora objetivos de calidad.
- Integra el desarrollo con el mantenimiento, etc.
- Además es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

### **3.3.2 DESVENTAJAS**

- Demostrar al cliente "exigente" (bajo contrato) que el enfoque evolutivo es controlable.
- Requiere gran habilidad y experiencia para valorar el riesgo y saber cuándo detener la evolución.
- Genera mucho tiempo en el desarrollo del sistema.
- Modelo costoso.

### **3.4 CONSTRUCCIÓN Y ASPECTOS METODOLÓGICOS**

Para la creación del sistema FAQ se hará uso de la metodología OMT y las técnicas de modelado de software Unified Modeling Language (UML), las cuales mediante el diseño y análisis de diagramas y modelos que decisiones permitan la construcción del software.

La construcción se realizó utilizando el lenguaje de programación Visual Studio 2010 bajo paradigma orientado a objetos, ampliamente utilizado casi en su totalidad. El programa Camtasia Estudio Recovery se crean los videos del sistema CGWEB; los videos están creados y comentados por opciones y sub opciones, además al ser una aplicación de escritorio se hará uso de la programación de dos capas, es decir, aplicación gráfica y gestión de base de datos; la primera capa creada con Visual Studio.Net 2010 y la segunda mediante SQL Server 2008 donde estarán guardados la base de datos del sistema CGWEB.

Las ayudas que se implementaron en la aplicación del sistema FAQ son justamente para dar soporte al usuario del sistema administrativo y comercial de la empresa.

Con los manuales existentes de cada módulo del sistema CGWEB se realizó videos debidamente comentados en cada uno de sus pasos lo que ayudara a que el cliente preste su conformidad con el sistema FAQ.

Para poder atender inmediatamente las preguntas que el usuario del sistema CGWEB realiza frecuentemente, se realizó un estudio de los manuales que tiene la empresa con una recopilación de los diferentes módulos del sistema.

Fue necesario una investigación preliminar y un proceso de preguntas con los usuarios del sistema en las que se puedo establecer las preguntas más frecuentes que realizan los usuarios

Con la ayuda del personal técnico de la empresa se facilitó la recopilación de toda la información archivada de las preguntas ya contestadas que los usuarios han venido realizando desde el año 2008, se realizó una clasificación en orden cronológico

### **3.5 METODOLOGÍA OMT**

(Object Modeling Technique) OMT está construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones.

OMT pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción, en el cual el análisis está enfocado en el mundo real para un nivel de diseño, también pone detalles particulares para modelado de recursos de la computadora. Esta Tecnología puede ser aplicada en varios aspectos de implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos.

OMT pone importancia en el uso de modelos para lograr una abstracción en el cual el análisis está enfocado en el mundo real para un nivel de diseño. La característica fundamental de OMT es tener en cuenta todos los principios del paradigma orientado a objetos y refuerza particularmente la necesidad de modelos consistentes de análisis y diseño del negocio como paso previo y fundamental al desarrollo de la programación y la definición de la base de datos. (<http://erubio.org>, 2013)

La virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software.

### **3.5.1 PROCESO DE DESARROLLO OMT**

Los pasos para desarrollar diseño OMT son:

#### ***3.5.1.1 Desarrollo***

Empieza con el análisis de la empresa o negocio, o de cómo los usuarios conciben el sistema y formulan sus requerimientos.

#### ***3.5.1.2 Conceptualización***

Es a menudo por la re ingeniería de procesos de la empresa, es una observación crítica de los procesos de la empresa, y su impacto económico.

En esta etapa se debe tener en cuenta las siguientes preguntas:

¿Cuál es la aplicación?

¿Qué problemas tendrán que ser resueltos?

¿Dónde será usado el sistema?

¿Cuándo será requerido el sistema?

¿Para qué es necesario el sistema?

### **3.5.2 FASES DE LA METODOLOGÍA OMT**

Las fases que conforman a la metodología OMT son:

#### ***3.5.2.1 Análisis***

Los requerimientos formados durante la conceptualización son revisados analizados para la construcción del modelo real. La meta del análisis es especificar las necesidades que deben ser satisfechas.

El analista construye un modelo del dominio del problema, mostrando sus propiedades más importantes. El modelo de análisis es una abstracción resumida y precisa de lo que debe de hacer el sistema deseado y no de la forma en que se hará. Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos tales como estructuras de datos. Un buen modelo debe poder ser entendido y criticado por expertos en el dominio del problema que no tengan conocimientos informáticos.

#### ***3.5.2.2 Diseño del sistema***

El diseño tiene un alto nivel estratégico y decisión para resolver los problemas. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta. Se selecciona una estrategia para afrontar el problema.

Los problemas grandes se deben ver desde el punto de análisis y diseño, este sistema se divide en subsistemas, a su vez este subsistema puede ser dividido en otros sub sistemas de manera que puedan ser manejados y cada componente pueda ser comprensible.

En esta etapa se deben crear estrategias, formular una arquitectura para el sistema y las políticas que deben guiarla además un detalle del diseño.

Se debe tener en cuenta los siguientes aspectos:

- Divisar una arquitectura.
- Elegir una implementación para un control externo.
- Si se usa base de datos elegir el paradigma de administración de base de datos.
- Determinar oportunidades para el reúso.
- Elegir estrategia para interacción de datos.

- Elegir una forma de identificar los objetos.
- Detallar el diseño.

### ***3.5.2.3 Diseño de objetos***

Este modelo describe la estructura estática de los objetos del sistema (identidad, relaciones con otros objetos, atributos y operaciones). El modelo de objetos proporciona el entorno esencial en el cual se pueden situar el modelo dinámico y el modelo funcional.

El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. OMT describe la forma en que el diseño puede ser implementado en distintos lenguajes (orientados y no orientados a objetos, bases de datos, etc.).

La programación orientada a objetos (POO) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación

Está basado en varias técnicas, incluyendo herencia cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Este tipo de programación se utilizó para hacer más sencilla la codificación de sistema para que luego sea reutilizable de tal modo que la programación del sistema fue sumamente clara y fácil para realizar alguna corrección

### ***3.5.2.4 Implementación***

Una implementación es la realización de una especificación técnica o algoritmos como un programa, componente software, u otro sistema de cómputo. Muchas implementaciones son dadas según a una especificación o un estándar

El diseño es trasladado a un lenguaje de programación actual y código de base de datos. Este paso puede ser aplicado y considerado durante el análisis y diseño

Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta. Durante la fase de implementación es importante tener en cuenta los principios de la ingeniería del software de forma que la correspondencia con el diseño sea directa y el sistema implementado sea flexible y extensible. No tiene sentido que utilicemos AOO y DOO de forma que potenciemos la reutilización de código y la correspondencia

entre el dominio del problema y el sistema informático, si luego perdemos todas estas ventajas con una implementación de mala calidad.

### 3.5.2.5 Mantenimiento

La documentación del desarrollo y seguimiento de los modelos a través del código facilita el posterior mantenimiento

Es una de las actividades más comunes en la ingeniería de software, es el proceso de mejora y optimización del software después de su entrega al usuario final (es decir; revisión del programa), así como también corrección y prevención de los defectos

### 3.5.3 METODOLOGÍA OMT CLASES PARA DESCRIBIR EL SISTEMA

En el lenguaje de modelado de software para desarrollar esta metodología del proceso es necesario trabajar algunos de los distintos diagramas que a continuación se enuncian, cada uno de estos diagramas aportan para el buen desarrollo del proceso final dando una visión general de los procesos y sus iteraciones, de tal forma que los programadores puedan realizar partes del sistema utilizando los pilares de la programación orientada a objetos como son la reutilización, la herencia y el polimorfismo.

<u>PARADIGMA ESPIRAL INCREMENTAL O.O</u>	<u>METODOLOGÍA OMT</u>	<u>TÉCNICAS DIAGRAMA UML</u>
<b>MODELAMIENTO Y DISEÑO DE DIAGRAMAS</b>		IDENTIFICACIÓN DE ACTORES DIAGRAMAS DE CASOS DE USO DICCIONARIO DE CASOS DE USO
<b>ANALISIS Y DISEÑO</b>	MODELO ESTÁTICO	DIAGRAMA DE CLASES DIAGRAMA DE OBJETOS DICCIONARIO DE CLASES
	MODELO DINÁMICO	DIAGRAMA DE INTERACCIÓN *Diagrama de Secuencias *Diagrama de Colaboración
	MODELO DE PROCESOS	DIAGRAMA DE ACTIVIDADES DIAGRAMA DE ESTADOS

Tabla 2 (Pressman, UML)

Modelamiento y diseño de diagramas en base a la metodología OMT con técnicas de modelado UML

### ***3.5.3.1 Identificación de actores***

Los actores representan un tipo de usuario del sistema. Entiendo como usuario cualquier cosa externa que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático o unidades organizativas o empresas.

Un actor en un diagrama de casos de uso representa un rol que alguien puede estar jugando, no un individuo en particular por lo tanto puede haber personas particulares que puedan estar usando el sistema de diferentes formas por ejemplo en el manejo de un software puede existir un usuario administrador y un usuario de lectura. (Manual técnico figura 1,2)

### ***3.5.3.2 Diagrama de casos de uso***

Muestra las relaciones entre actores y casos de uso del sistema, un caso de uso representa una funcionalidad del sistema, la cual puede interactuar con elementos externos como actores u otros sistemas, nos da el paso a paso cada una de las funciones que va a realizar el programa de una forma modular, de tal forma que se exprese de una forma gráfica todo lo que va a hacer el sistema. Se puede decir que es base de muchos de los otros diagramas que se utilizan en UML, ya que muestra que se pretende hacer para suplir la necesidad que se va a solucionar con el sistema. (Manual técnico figura 3,4)

### ***3.5.3.3 Diccionario de casos de uso***

Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. En los diagramas de caso de uso se modelan la funcionalidad del sistema usando actores y casos de uso los cuales son detallados en el diccionario como servicios y funciones provistas por el sistema para los usuarios. (Manual técnico figura 6,7)

## **3.6.1 MODELO ESTÁTICO**

Describe la estructura estática de los objetos del sistema (identidad, relaciones con otros objetos, atributos y operaciones). El modelo de objetos proporciona el entorno esencial en el cual se pueden situar el modelo dinámico y el modelo

funcional. El objetivo es capturar aquellos conceptos del mundo real que sean importantes para la aplicación. Se representa mediante diagramas de objetos. (Manual técnico figura 8, 9,10)

#### ***3.6.1.1 Diagrama de clases***

Este diagrama nos muestra cada una de las clases, entendiéndose por clase la parte funcional de un sistema, de igual forma puede generalizarse como un objeto, se puede decir que es la columna vertebral del sistema, una clase está compuesta por un nombre que la identifica, atributos que son necesarios para su función y métodos que son las funciones propias que desempeña. Las clases se relacionan entre sí para conformar un sistema orientado a objetos. Este diagrama se utiliza en el proceso de análisis y diseño donde se crea un diseño conceptual para ver cada uno de los componentes que se encargaran del funcionamiento.

Los diagramas son 13 pero en realidad estos se dividen llegando hacer solamente 6 diagramas los que se utilizan, su uso va de acuerdo a las necesidades propias de cada sistema, lo ideal es lograr una buena explicación del sistema gráficamente expresada. (Manual técnico figura 10)

#### ***3.6.1.2. Diagrama de objetos***

Representan un único ejemplo de una clase y se utilizan para ilustrar un punto de datos en su aplicación. Cuando cree un objeto nuevo, llamado especificación de instancia, le permite asignar una clase ya existente representada por la instancia.

Los diagramas de objetos pueden ayudar a explicar las clases y su herencia. A veces se dibujan durante el proceso de planificación de clases o para ayudar a partes interesadas para quienes los diagramas de clases sean demasiado abstractos. (Manual técnico figura 11)

#### ***3.6.1.3 Diccionario de clases***

El diccionario de datos de las clases simplemente detalla cada una de las clases que se han descubierto, se muestra el nombre de la clase (que por convención se hace con un 'C\_' y el nombre con mayúsculas en sus letras iniciales y sin espacios), sus atributos y sus operaciones. No se da aún un tipo de dato para los atributos pues esto es parte del diseño. (Manual técnico pag 8)

### **3.7.1 MODELO DINÁMICO**

Describe los aspectos de un sistema que tratan de la temporización y secuencia de operaciones (sucesos que marcan los cambios, secuencias de sucesos, estados que definen el contexto para los sucesos) y la organización de sucesos y estados. Captura el control, aquel aspecto de un sistema que describe las

secuencias de operaciones que se producen sin tener en cuenta lo que hagan las operaciones, aquello a lo que afectan o la forma en que están implementadas. Se representa gráficamente mediante diagramas de estado. (Manual técnico figura 12, 13,14)

#### ***3.7.1.1 Diagrama de interacción***

Representa la forma como el actor (usuario) y los objetos se comunican entre sí a la petición de un evento, en este tipo de diagramas se encuentran dos que se encargan de aspectos particulares, diagrama de secuencia y diagrama de colaboración. (Manual técnico figura 12)

#### ***3.7.1.2 Diagrama de secuencia***

Este nos muestra la interacción ordenada a través del tiempo de cada evento, muestra los objetos participantes en cada interacción y los mensajes que intercambian según su secuencia. En él se puede visualizar lo que pasa con el sistema a través del tiempo y se pueden dar las pautas que se tienen cuando es necesario designar un tiempo límite para los procesos, de igual forma se pueden ver los mensajes que viajan entre cada uno de los procesos. (Manual técnico figura 13)

#### ***3.7.1.3 Diagrama de colaboración***

Muestra la interacción entre cada objeto, a diferencia del diagrama de secuencia nos muestra la relación entre cada evento y objeto, es utilizado sobre todo cuando es necesario incluir elementos externos habitualmente hardware que interactúa con el sistema, de donde se toman datos o a donde se exportan datos con un fin específico.

### **3.8.1 MODELO DE PROCESOS**

Describe las transformaciones de valores de datos (funciones, correspondencias, restricciones y dependencias funcionales) que ocurren dentro del sistema. Captura lo que hace el sistema, independientemente de cuando se haga o de la forma en que se haga. Se representa mediante diagramas de flujo de datos. (Manual técnico figura 14, 15, 16)

#### ***3.8.1.1 Diagrama de actividades***

Representa la serie de actividades realizadas en cada caso de uso como también las diferentes rutas que puede desencadenar cada evento en este diagrama utilizamos lo siguiente:

Inicio: Donde tiene inicio el diagrama de actividades representado por un círculo de color negro sólido.

Actividad: Acción realizada por el sistema representado con un ovalo. (Manual técnico figura 14,15)

### **3.8.1.2 Diagrama de estados**

Muestra la secuencia de estados por las que pasa un caso de uso, un objeto a lo largo de su vida en este se muestra cada evento que cambia de estado en todo el sistema (documentar). (Manual técnico figura 16)

## **3.9 LENGUAJE UNIFICADO DE MODELADO UML**

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas software más conocido y utilizado en la actualidad

UML es el lenguaje estándar para realizar el modelado de los sistemas de software independiente del lenguaje de programación que se utilice.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos. (Pressman, *Lenguaje de modelado UML*, 2010)

### **3.9.1 OBJETIVO UML**

Visualizar, Especificar, Construir, Documentar mostrando la manera en la que un usuario final va a interactuar con el sistema a desarrollar, sin preocuparse por la forma en la que se va a lograr implementar eso, técnicamente hablando, es decir, sin tomar en cuenta los mecanismos que se van a utilizar para crear o hacer funcionar el sistema.

## **CONCLUSIONES**

El presente proyecto ha contribuido con la empresa IT del Ecuador optimizando el proceso de soporte funcional al cliente donde se atiende inmediatamente a los usuarios que usan el sistema CGWEB reduciendo el tiempo de espera, para responder a las preguntas generadas por los usuarios.

La información del sistema FAQ se mantendrá en un nivel de privacidad, al que solo podrán acceder los usuarios del sistema con cada uno de los roles asignados por la empresa .

Mediante el estudio de las especificaciones técnicas en base al modelo vista controlador se logró la creación de un sistema con un ambiente amigable, robusto y seguro para el manejo de los usuarios.

La implantación de videos en los servidores se la hizo cuidando el tiempo de carga, calidad, pertinencia de contenidos, donde se explica paso a paso el funcionamiento del sistema CGWEB, aclarando cualquier duda que al usuario se le presente en su trabajo diario.

## **RECOMENDACIONES**

- Se recomienda que la persona administrador de la empresa IT del Ecuador establezca políticas de seguridad para realizar alguna actualización del sistema FAQ.
- El responsable de usar el sistema como usuario administrador debe tener conocimientos necesarios con el manejo de base de datos para que este en la capacidad de realizar actualizaciones o a la vez dar el mantenimiento adecuado
- En el caso de presentarse errores al ingreso del sistema FAQ deben acudir al usuario administrador que tendrá los accesos correspondientes para solucionar los problemas.
- Se debe ir realizando encuestas a los usuarios donde se manifieste la satisfacción que tienen con el sistema FAQ, tomando sugerencias para mejoramiento continuo y actualización de las preguntas frecuentes que se les vaya presentando y no existan en la base de datos.

## BIBLIOGRAFÍA

### RECOMENDACIONES WEB

- <http://erubio.org>. (2013). Recuperado el 10 de 08 de 2013, de Metodología OMT: <http://erubio.org/menchaca/dbdd/cap7/sld021.htm>
- <http://www.fismat.umich.mx>. (2013). Recuperado el 05 de 10 de 2013, de Interfaz de usuario: <http://www.fismat.umich.mx/~crivera/tesis/node6.html>
- <http://www.maestrosdelweb.com>. (2013). Recuperado el 10 de 02 de 2013, de Maestros del web: <http://www.maestrosdelweb.com/editorial/web2/>
- <http://www.microsoft.com>. (2013). Recuperado el 10 de 07 de 2013, de Lenguajes de programación: <http://www.microsoft.com/es-es/download/details.aspx?id=21>
- <http://www.ri5.com.ar/ayuda03.php>. (2013). Recuperado el 14 de 05 de 2013, de Lenguaje HTML: <http://www.ri5.com.ar/ayuda03.php>
- <http://www.slideshare.net>. (2013). Recuperado el 01 de 08 de 2013, de Introducción a la Plataforma .NET: <http://www.slideshare.net/jnarchie/introduccion-a-la-plataforma-net>
- <http://www.unalmed.edu.com>. (2013). Recuperado el 12 de 08 de 2013, de Ing de Software: [http://www.unalmed.edu.co/~mstabare/disenio\\_conceptual.htm](http://www.unalmed.edu.co/~mstabare/disenio_conceptual.htm)
- <http://www.visualstudio.com/>. (2013). Recuperado el 22 de 08 de 2013, de Visual Studio: <http://www.visualstudio.com/>
- <http://www.w3schools.com>. (2013). Recuperado el 10 de 07 de 2013, de HTML 5 : [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- <http://www.yiiframework.com>. (2013). Recuperado el 10 de 08 de 2013, de Mejores prácticas MVC: <http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices>
- <http://technet.microsoft.com>. (15 de 12 de 2013). *tráfico de Internet*. Obtenido de <http://technet.microsoft.com/en-us/library/ee809069.aspx>

## LIBROS

- Pressman. (2010). Características de Software. En Pressman, *Ing de Software* (págs. 3,4). Edición 7.
- Pressman. (2010). Dominios de Aplicación de Software. En Pressman, *Ing Software* (págs. 6,7). Edición.
- Pressman. (2010). Lenguaje de modelado UML. En Pressman, *Modelado de los requerimientos* (pág. 137). Edición 7.
- Pressman. (2010). Modelo de procesos. En Pressman, *Proceso de software* (págs. 38,39). Edición 7.
- Pressman. (2010). Modelo de procesos. En Pressman, *El proceso de Software* (pág. 48). Edición 7 .
- Pressman. (2010). Proceso del Software. En Pressman, *Ing del software* (págs. 12,13). Edición 7.
- Pressman. (7ma Edición). *Modelo Espiral*.
- Pressman. (s.f.). *UML*.
- Pressman, R. (2010). Ing Software. En R. Pressman, *Ing de Sotware* (págs. 11,12). Edición 7.
- Pressman, R. (2010). Un enfoque practico. En R. Pressman, *Ing de Software* (págs. 1,2). Edición 7.
- Sznajdleder, P. (2010). Estudio del lenguaje y desarrollo de aplicaciones. En P. Sznajdleder.
- Sznajdleder, P. (2010). Estudio del lenguaje y desarrollo de aplicaciones. En P. Sznajdleder.