



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“SISTEMA PARA LA ADMINISTRACIÓN DE UNIDADES DE  
POSTGRADO”**

**TESINA DE SEMINARIO**

Previo a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES  
ESPECIALIZACIÓN EN SISTEMAS TECNOLÓGICOS**

Autores:

**DIANA NATHALY LÓPEZ ARMENDÁRIZ  
WALTER EUSEBIO ANTÓN ESPINOZA**

**GUAYAQUIL – ECUADOR**

**Año: 2010**

# **AGRADECIMIENTO**

Agradecemos a nuestro Dios por habernos permitido culminar satisfactoriamente y con éxitos nuestras metas profesionales y dado fortaleza y la sabiduría necesaria.

A nuestros amigos por el apoyo incondicional en buenos y malos momentos.

Un agradecimiento especial al Ingeniero Carlos Martin y al Ingeniero Lenin Freire por habernos guiado con sus conocimientos y consejos para hacer posible la culminación de nuestro Proyecto de Grado.

## **DEDICATORIA**

A mi Señor, Jesús, quien me dio la fe, la fortaleza, la salud y la esperanza para terminar este trabajo.

A mis padres por todo lo que me han dado en esta vida, ya que gracias a ellos soy quien soy hoy en día, fueron los que me dieron ese cariño, calor humano necesario, han velado por mi salud, mis estudios, por ser la fuente de mi inspiración y motivación para superarme cada día más.

A mi hermano por su ayuda, comprensión y estar siempre conmigo.

**Diana López Armendáriz**

Les dedico este proyecto a Dios y su santa madre la Virgen María quien me acogió bajo su manto en toda mi vida de colegio y me ha protegido en mi vida Universitaria.

A mis padres por todo su apoyo y por haberme obsequiado el mejor regalo que un hijo puede recibir de ellos, mis estudios universitarios. A mi abuelita que desde muy pequeño me supo aconsejar y desde donde se encuentra aun me sigue cuidando. A mis hermanos por su apoyo y confianza en la culminación de mi carrera.

**Walter Antón Espinoza**

# **TRIBUNAL DE SUSTENTACIÓN**

---

MSC. Carlos Martin

**PROFESOR DEL SEMINARIO**

---

MSIG. Lenin Freire

**PROFESOR DELEGADO DEL DECANO**

# DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**”

(Reglamento de exámenes y títulos profesionales de la ESPOL).

---

Diana Nathaly López Armendáriz

---

Walter Eusebio Antón Espinoza

# RESUMEN

El presente trabajo consiste en la implementación de un Sistema de Administración de Unidades de Postgrados que permita el registro de los estudiantes que desean tener estudios de especialización que se cursa tras la graduación o licenciatura. Se trata de la formación que sigue al título de grado, que es la titulación de educación superior que se obtiene al concluir una carrera universitaria.

El Sistema es una aplicación la cual permite la administración como son ingreso de nuevos estudiantes con toda la información necesaria, así también se presentará las tesis y tutorías que los alumnos tiene asignados, información sobre los pagos que los estudiantes realizan durante el proceso de Postgrado.

# INDICE GENERAL

RESUMEN.....	I
INDICE GENERAL.....	II
ANEXOS.....	IV
ABREVIATURAS.....	V
INDICE DE FIGURAS Y GRÁFICOS.....	VI
INDICE DE TABLAS.....	VII
INTRODUCCIÓN.....	VIII
CAPÍTULO I.....	1
1. ANÁLISIS CONTEXTUAL.....	1
1.1. Antecedentes.....	1
1.2. Justificación.....	2
1.3. Objetivos del proyecto.....	3
1.3.1. Objetivos Generales.....	3
1.3.2. Objetivos Específicos.....	4
CAPÍTULO II.....	6
2. FUNDAMENTACIÓN TEORICA.....	6
2.1. Desarrollo de Aplicaciones Web.....	6
2.2. Descripción General de AJAX.....	7
2.3. Ventajas de AJAX.....	12
2.4. Arquitectura Modelo Vista Controlador.....	14
2.4.1. Descripción del patrón.....	15
2.4.2. Características del Modelo Vista Controlador.....	16
CAPÍTULO III.....	18
3. ANÁLISIS DE IMPLEMENTACIÓN.....	18
3.1. Estudio de viabilidad.....	18
3.1.1. Viabilidad económica.....	19
3.1.2. Viabilidad técnica.....	20

3.2. Análisis Económico y Técnico .....	21
3.2.1. Calidad .....	22
3.2.2. Costo .....	22
CAPÍTULO IV .....	23
4. ESPECIFICACIONES Y HERRAMIENTAS PARA LA SOLUCIÓN .....	23
4.1. Especificaciones Técnicas .....	23
4.1.1. Requerimientos en Hardware .....	23
4.1.2. Requerimientos en Software .....	24
4.2. Herramientas basadas en Software .....	25
4.2.1. Ambiente de desarrollo (IDEs) .....	25
4.2.2. Librerías y componentes .....	25
CAPÍTULO IV .....	27
5. DISEÑO Y METODOLOGIA UTILIZADA .....	27
5.1. Aplicando arquitectura MVC .....	27
5.2. Modelo lógico de la base de datos .....	27
5.3. Módulos del Sistema .....	30
5.3.1. Definición de roles en los módulos .....	32
5.4. Casos de Estudio .....	34
5.4.1. Especificación de Actores .....	34
CONCLUSIONES Y RECOMENDACIONES .....	44
APENDICE A: .....	46
DISEÑO DE LA BASE DE DATOS (DDL) .....	46
APENDICE B: .....	57
PRUEBAS DE USABILIDAD .....	57
ENCUESTAS DE EVALUACIÓN A LOS USUARIOS .....	65
GLOSARIO .....	69
REFERENCIAS BIBLIOGRÁFICAS .....	71



# **ANEXOS**

**1. ANEXO A: DISEÑO DE LA BASE DE DATOS (DDL)**

**2. ANEXO B: PRUEBAS DE USABILIDAD**

# ABREVIATURAS

Las abreviaturas presentadas en la siguiente tesis son las siguientes:

<b>AJAX:</b>	Asynchronous JavaScript And XML
<b>XML:</b>	Extensible Markup Language
<b>API:</b>	Application Programming Interface
<b>CSS:</b>	Cascading Style Sheets
<b>DOM:</b>	Document Object Model
<b>HTTP:</b>	Hypertext Transfer Protocol
<b>JS:</b>	Java Script
<b>MVC:</b>	Modelo Vista Controlador
<b>SQL:</b>	Structured Query Language
<b>URL:</b>	Uniform Resource Locator
<b>XHTML:</b>	Hypertext Markup Language

## INDICE DE FIGURAS Y GRÁFICOS

<b>Figura 1.1</b> Modelo conceptual del SAP.....	5
<b>Figura 2.1</b> El patrón de interacción sincrónica y patrón asincrónico de una aplicación AJAX.....	8
<b>Figura 2.2</b> Representación del modelo clásico y modelo ajax. ....	10
<b>Figura 2.3</b> Secuencia clásica sin Ajax – Secuencia clásica con Ajax. ....	11
<b>Figura 2.4</b> Esquema Modelo Vista Controlador .....	15
<b>Figura 2.5</b> Funcionamiento Modelo Vista Controlador .....	17
<b>Figura 5.1:</b> Modelo lógico de la base de datos. ....	28
<b>Figura 5.2</b> Módulos del Sistema.....	31

## INDICE DE TABLAS

<b>Tabla 3.1</b> Costos del Proyecto .....	19
<b>Tabla 3.2</b> Requerimientos por grado de dificultad.....	21
<b>Tabla 4.1</b> Requerimientos por grado de dificultad.....	24
<b>Tabla 4.2</b> Componentes basados en Software.....	24
<b>Tabla 5.1.</b> Definición de Roles en los Módulos.....	33

# INTRODUCCIÓN

La Unidad de Postgrado, es un centro que genera, difunde, aplica el conocimiento, habilidades, a través de la docencia y la investigación. Esta dirigido para personas nacionales y extranjeras escogidas a través de un riguroso proceso de admisión, cuyo propósito es garantizar que los futuros estudiantes tengan la capacidad, motivación, y madurez necesaria para culminar con éxito.

Personas que son profesionales técnicos en las áreas de Ingeniería de Sistemas y Telecomunicaciones, conscientes que el potencial de la tecnología por si mismo no servirá para lograr resultados positivos en las organizaciones, sin un liderazgo y habilidades gerenciales que guíen su implantación, planeación, diseño y administración como apoyo a los procesos de su organización.

Este proceso se maneja a través de ciertas tareas que los usuarios deben analizar y verificar. Toda la información ingresada es verificada por el Consejo de Postgrado.

En la actualidad se maneja un control a través de carpetas en las cuales se almacena los documentos que el estudiante presenta al momento del

registro, algunos de ellos llenados a mano, por la acumulación de carpetas se están dañando, deteriorando e incluso se pueden perder, lo que causara problemas para la Unidad de Postgrados al momento de realizar algún control o seguimiento de los estudiantes.

Para lo cual el sistema presentará funciones útiles las cuales permitirán a los usuarios en general interactuar con todos los procesos que participan en el control de la unidad de Postgrados, la cual facilitara las tareas para la selección de los futuros estudiantes.

Propósito en cual se llega automatizar las tareas que se llevan a cabo dentro de la Unidad de Postgrados, para así gestionar de manera eficiente el uso y manipulación de datos.

# **CAPÍTULO I**

## **1. ANÁLISIS CONTEXTUAL**

### **1.1. Antecedentes**

La Unidad de Postgrados que se encarga de brindar como servicio la formación de recursos humanos de alta calidad a nivel de Postgrado, para lo cual cuentan con los programas de Maestría en Sistemas de Información y Maestría en Seguridad Informática Aplicada, realizan un proceso de selección de los futuros estudiantes.

En la actualidad se utilizan carpetas en las cuales se almacena los documentos que el estudiante presenta al momento del registro, algunos de ellos llenados a mano, por la acumulación de carpetas se están dañando, deteriorando e incluso se pueden perder, lo que causara problemas para la unidad de Postgrados al momento de realizar algún

control o seguimiento de los estudiantes.

Para el usuario el ingreso de información es muy importante y con el proceso actual le resulta demasiado incomodo puesto que no garantiza encontrar la carpeta del estudiante.

Actualmente existen dos personas las cuales realizan este tipo de trabajo ellos son los más afectados con respecto al tiempo que toma en realizar las tareas, aunque también existe un tercer afectado que es el estudiante puesto que no existe una página en la cual él pueda realizar consultas de los pagos, aprobación de tesis y el servicio de tutoría que ofrece la unidad de Postgrados.

## **1.2. Justificación**

Las grandes empresas en la actualidad se hacen más competitivas dentro de su rama y cada vez adoptan más estrategias a fin de garantizar el éxito. La necesidad de plantear la evaluación de los procesos, incidirá en la optimización de los mismos.

El Sistema cubrirá las necesidades de corregir estos procesos como son el ingreso de estudiantes, docentes, seguimiento de interesados, las consultas de los pagos que tiene un estudiante, consultas acerca de posibles descuentos de acuerdo a ciertas condiciones.



### **1.3. Objetivos del proyecto**

Al desarrollar el sistema se quiere cumplir con los siguientes objetivos:

#### **1.3.1. Objetivos Generales**

Diseñar e implementar un sistema que permita administrar el ingreso de nuevos estudiantes y guardar toda la información la necesaria.

Implementar un sitio web que sirva de información para que los estudiantes puedan conocer como se encuentran sus pagos.

Implementar una sección en el cual los estudiantes puedan visualizar la información sobre el servicio de tutorías que ofrece la Unidad de postgrado.

Combinar una serie de modelos, métodos, componentes y herramientas para el análisis de la distribución de los datos. Estas soluciones están directamente relacionadas con la bases de datos su diseño, administración y recuperación de la información.

Contribuir a la reducción de tiempo del manejo de tareas y proporcionar a los usuarios un mejor control de la actividades que se llevan en esta entidad con en el fin de automatizar los procesos que se efectúan.

### **1.3.2. Objetivos Específicos**

Para alcanzar los objetivos generales, se deben completar los siguientes objetivos específicos:

- Creación de una interfaz web para la administración, ingreso de nuevos estudiantes con toda la información necesaria.
- Implementar una sección donde el administrador pueda ingresar los datos de Postgrado.
- Implementar una sección donde el administrador pueda ingresar una promoción, para así poder asignar a todos los Postgrados.
- Implementar la creación de tutorías, servicio que ofrece la Unidad de Postgrados.
- Implementar una sección en la cual se presenta la tesis que tiene asignada un alumno.
- Diseñar una interfaz para que los estudiantes puedan visualizar los pagos que han realizado en durante el proceso del Postgrado.

- Diseñar la interfaz en el cual se pueda presentar un seguimiento de llamadas, en cual presenta las personas interesadas en el servicio que ofrece la Unidad de Postgrado.



**Figura 1.1** Modelo conceptual del sistema de Administración de Unidades de Postgrados.

## **CAPÍTULO II**

### **2. FUNDAMENTACIÓN TEORICA**

#### **2.1. Desarrollo de Aplicaciones Web**

El desarrollo de Aplicaciones web requiere de factores muy importantes como son navegabilidad, interactividad, usabilidad, arquitectura de información y la interacción de los medios. [2]

Factores que han evolucionado debido a las herramientas de desarrollo, por lo cual es importante mencionar que una aplicación web puede poseer elementos, los cuales permiten la comunicación entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, se puede realizar este tipo de manipulación debido a que la página puede responder a las acciones, como por ejemplo llenar y enviar

formularios es decir accedemos a la base de datos. Pero hay ciertas aplicaciones web las cuales mantienen restricciones, es decir no permite la manipulación del estado de la información.

Se debe realizar un análisis de las necesidades, para obtener un buen diseño con una jerarquía bien elaborada de contenidos para así aumentar la eficiencia de la aplicación web.

## **2.2. Descripción General de AJAX**

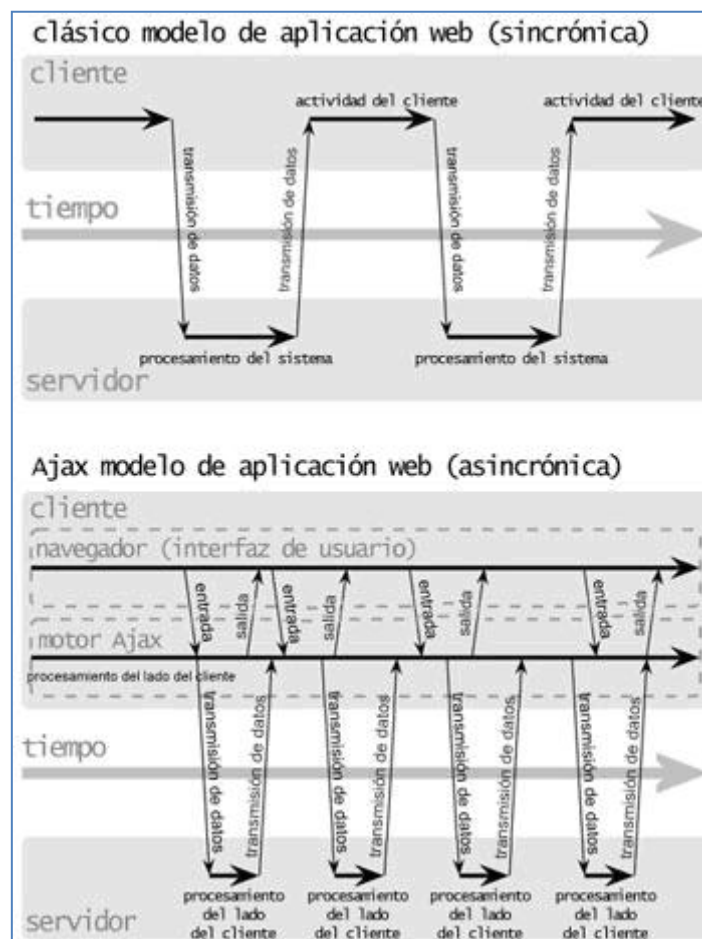
Ajax es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. [10]

### **¿Porque es tan interesante AJAX?**

Porque en realidad AJAX no es una tecnología, sino la unión de varias tecnologías que juntas pueden lograr cosas realmente impresionantes como GoogleMaps, Gmail o algunas otras aplicaciones muy conocidas: AJAX, en resumen, es el acrónimo para Asynchronous JavaScript + XML y el concepto es: Cargar y re-direccionar una página, luego mantenerse

en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-direccionando la página y mostrando u ocultando porciones de la misma.

La figura 2.1 representa El patrón de interacción sincrónica de una aplicación Web tradicional (arriba) comparada con el patrón asincrónico de una aplicación AJAX (abajo).



**Figura 2.1** El patrón de interacción sincrónica y patrón asincrónico de una aplicación AJAX.

## Definiendo AJAX

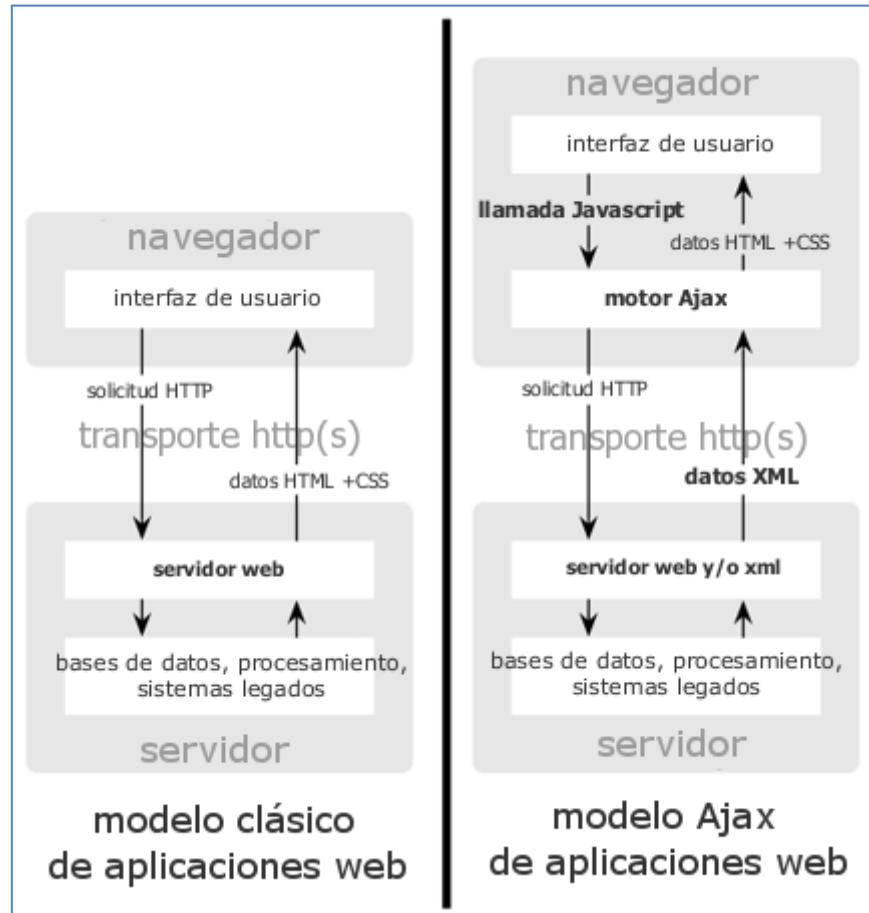
Ajax no es una tecnología, es la suma de técnicas y tecnologías que se comparten para partir el contenido y la programación y dialogar con el servidor mientras la página se refresca. AJAX incorpora:

- Document Object Model (DOM) para visualizar dinámicamente e interactuar con la información presentada.
- Intercambio y manipulación de datos usando XML.
- XMLHttpRequest para recuperar datos de forma asincrónica.
- JavaScript como nexo de unión de todas las tecnologías

Ajax es un patrón de diseño que propone un nuevo modelo de interacción web combinando las tecnologías anteriores.

El modelo clásico de aplicaciones Web funciona de esta forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una pagina HTML al cliente.

En la siguiente figura 2.2 se representa el modelo tradicional para las aplicaciones Web (izquierda.) comparado con el modelo de AJAX (derecha.).



**Figura 2.2** Representación del modelo clásico y modelo ajax.

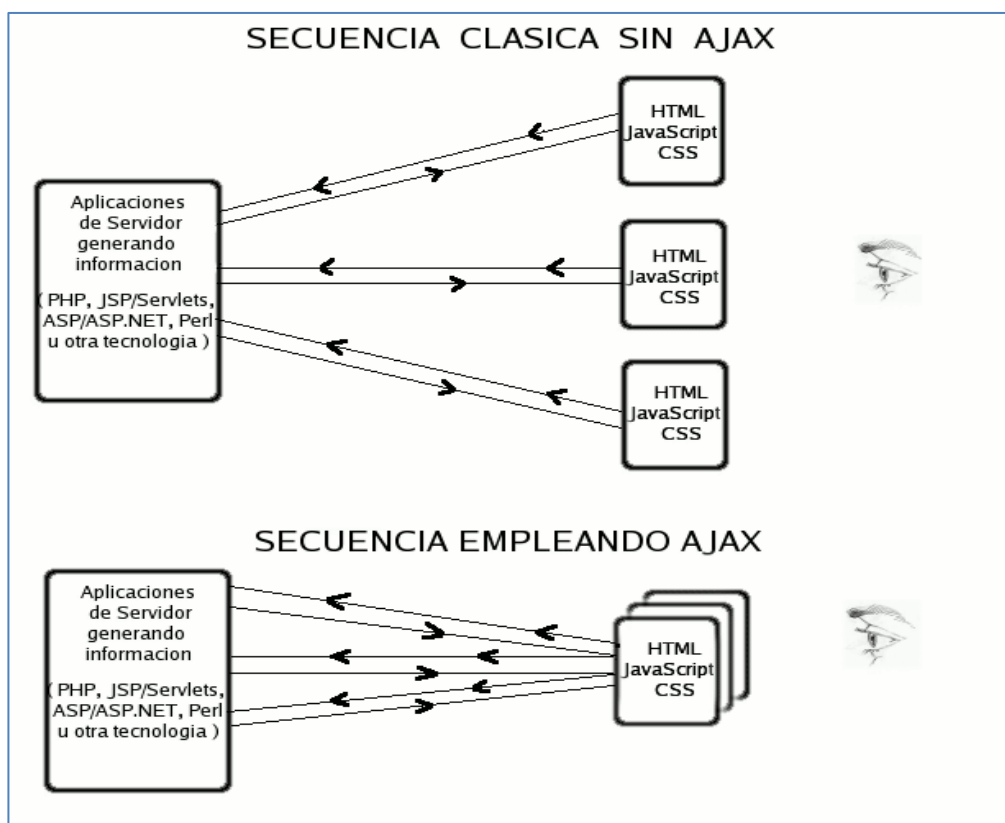
### Como es diferente AJAX

Una aplicación **AJAX** elimina la naturaleza “arrancar-frenar- arrancar-frenar” de la interacción en la Web introduciendo un intermediario -un motor **AJAX**- entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo



contrario. En vez de cargar un pagina Web, al inicio de la sesión, el navegador carga al motor **AJAX** (escrito en JavaScript).

Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario. Permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo. [10]



**Figura 2.3** Secuencia clásica sin Ajax – Secuencia clásica con Ajax.

### **2.3. Ventajas de AJAX**

Las ventajas del Ajax son muchas, citaremos algunas y daremos una explicación con una explicación más detallada.

Una de las ventajas del Ajax es que hace más rápida las operaciones, con esto estamos más cerca de crear aplicaciones web, damos menos carga del servidor ósea menos transferencia, menos ancho de banda (nos podemos ahorrar mucho dinero si tenemos muchas visitas).

Con el Ajax es hacer más cómoda la navegación y la interacción usuario-servidor. La experiencia de usuario en la navegación es mucho más interactiva. Ya no se refresca la página constantemente al interactuar con ella y el tiempo de espera para una petición se reduce ya que el usuario al hacer un pedido (request) al servidor, no se envía toda la página.

Las demandas puestas sobre las aplicaciones habilitadas en ambientes "Web" han venido incrementándose a lo largo de los años. Hoy en día, la técnica más reciente para hacer frente a estas necesidades es AJAX o "Asynchronous JavaScript and XML".

A pesar de la gran gamma de tecnologías utilizadas en aplicaciones empleadas en Internet, como: HTML/XHTML, CSS, JavaScript, Flash y XML, la característica primordial de todas estas tecnologías radica en que

su uso en un documento debe ser realizado en diversas requisiciones de navegación o "Reloads".

Ejemplifiquemos una secuencia típica de navegación para consultar un catálogo de productos a través de una aplicación web:

- Usuario selecciona criterio de búsqueda a través de una forma HTML/XHTML.
- Usuario obtiene resultados de búsqueda.
- Usuario refina criterio de búsqueda con resultados obtenidos.

Esta simple secuencia en la mayoría de aplicaciones existentes hoy en día, conlleva que el usuario observe documentos totalmente regenerados en cada ocasión, esto se debe a que la información desplegada en cada paso requiere obtener los datos de una aplicación en servidor, sea PHP, JSP/Servlets, ASP/ASP.NET, Perl u otra tecnología de servidor.

El proceso de obtener la información a través de una aplicación de servidor es inevitable, sin embargo, el mecanismo de regenerar el documento que visualiza el usuario en cada requisición, puede ser evitado empleando AJAX.

Al tener la capacidad de obtener información para actualizar un mismo documento, sin que el Navegador modifique su estado, el usuario experimenta una interfaz gráfica sumamente responsiva, ya que no pierde

de vista la información original del documento, sino ésta es actualizada en ciertas secciones según lo haga en su proceso de navegación.

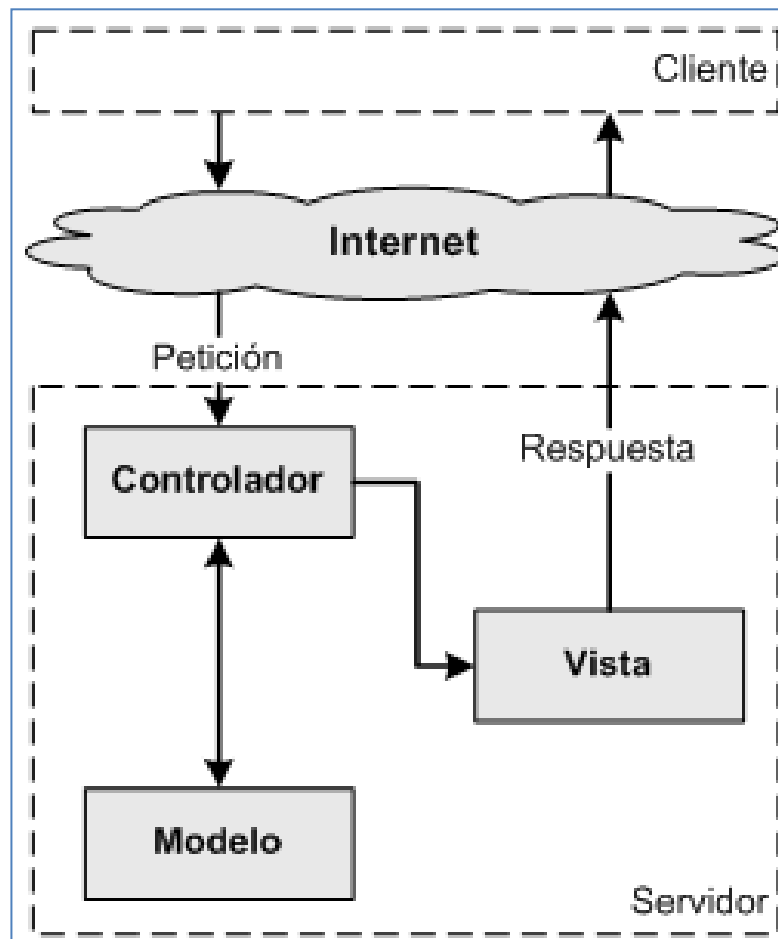
La información actualizada continua proviniendo de diversos URL's -- direcciones de Internet -- que poseen un comportamiento idéntico a cualquier aplicación de servidor, como si estas fuesen llamadas de manera independiente de AJAX.

## **2.4. Arquitectura Modelo Vista Controlador**

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

El estilo de llamada y retorno se visualiza 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. [12]

En la figura 2.4 se representa el esquema Modelo Vista Controlador.



**Figura 2.4** Esquema Modelo Vista Controlador

### 2.4.1. Descripción del patrón

La arquitectura MVC fue diseñada para reducir el esfuerzo al momento de programar. Además porque permite una fácil separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.

Este modelo de arquitectura presenta varias ventajas:

- Fácil separación entre los componentes de un programa; lo cual nos permite implementarlos por separado.
- API bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a nuestro diseño, las piezas de un programa se pueden construir por separado y luego unir las en tiempo de ejecución. Si uno de los componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. Todos tienen un Frame que contiene todos los elementos, un controlador de eventos, un montón de cálculos y la presentación del resultado. Ante esta perspectiva, hacer un cambio aquí no es nada trivial.

#### **2.4.2. Características del Modelo Vista Controlador**

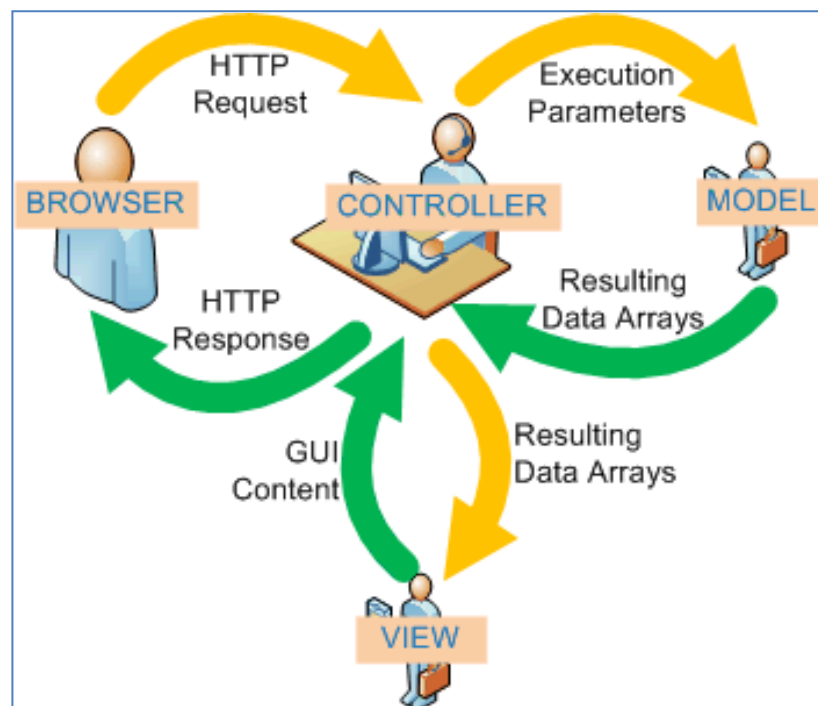
**Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas.

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:** Transforma el modelo en una página web permite al usuario interactuar con ella.

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

En la siguiente figura 2.5 se presenta el funcionamiento Modelo Vista Controlador.



**Figura 2.5** Funcionamiento Modelo Vista Controlador

## **CAPÍTULO III**

### **3. ANÁLISIS DE IMPLEMENTACIÓN**

#### **3.1. Estudio de viabilidad**

Muchas de las veces cuando se implementa un proyecto de software una de las principales inquietudes que tiene un grupo de desarrollo es si la aplicación es alcanzable tanto en la lógica de programación como en el coste total del proyecto. Este alcance es lo que permite decidir si continuar o no con el desarrollo del mismo.

La viabilidad y el análisis de riesgo están relacionados de muchas maneras tal y como se explica en la siguiente referencia. **[1]** Por lo tanto la evaluación del proyecto como tal comparando los beneficios contra los costos de producción es la normativa que indicará que tan viable es el proyecto, para ello se analizará tres puntos fundamentales que servirán de guía para la implementación del mismo.



### 3.1.1. Viabilidad económica

Un análisis económico para la iniciación de todo proyecto es necesario dado que un gasto elevado que podría incurrir en la solución en general. Gastos como licencias de productos que se necesiten adquirir para el desarrollo de la implementación.

Para ello en la Tabla 2.1 se menciona los costes de ciertos productos usados.

No	Producto	Costo
1	Windows Server 2003	Entre \$469 y \$500
2	SQL Server 2008 Developer Edition	Entre \$50 y \$60
3	Servidor Web	Entre \$600 y \$1200
	<b>TOTAL</b>	Entre \$1020 y \$1760

**Tabla 3.1** Costos del Proyecto

La tabla anterior se muestra los costes básicos para implementar la solución y hacer uso de la misma. Entre ellas mencionamos:

- *Windows Server 2003*
  - Sistema operativo orientado a servidores web. Más detalles en [3].

- *SQL Server 2008 Developer Edition*
  - Base de Datos. Para las pruebas se usó la versión Express pero es recomendable hacer uso de una licencia debido a que las versiones Express son gratuitas y por ende tienen un máximo de 4Gb para almacenar información. Más detalles en [4].
  
- *Servidor Web*
  - Computador con capacidad en hardware para ser usado como un servidor de aplicaciones.

### **3.1.2. Viabilidad técnica**

Esto implica preguntas por ejemplo si el desarrollo aplicando la tecnología es técnicamente alcanzable, es decir si los requerimientos son posibles de realizar. El estudio se basa en un diseño del contorno de los requisitos del sistema en términos de entrada, procesos, salida, campos, programas, y procedimientos. Esto se puede calificar en niveles de dificultad.

En la siguiente Tabla 3.1 se presenta por grado de dificultad, para calificar los requerimientos más importantes.

N.	Requerimiento	Grado de Dificultad		
		Básico	Medio	Avanzado
1	Ingreso de estudiantes		x	
2	Ingreso de postgrados y promoción.		x	
3	Creación y administración de tutorías.		x	
4	Creación y administración de tesis.		x	
5	Ingreso del Seguimiento de llamadas.		x	
6	Ingreso de Pagos			x
7	Administración del proceso de pagos			x

**Tabla 3.2** Requerimientos por grado de dificultad.

### 3.2. Análisis Económico y Técnico

El análisis Económico y Técnico incluye lo que se conoce como el análisis de costo y calidad que significa una comparación de los gastos realizados en la inversión de un producto y los beneficios que pueda obtener con el uso del mismo.

### **3.2.1. Calidad**

Para que la aplicación tenga un correcto funcionamiento esta deberá tener márgenes de calidad. A continuación se analizarán los más importantes.

- **Integridad y seguridad.** Para que el sistema sea seguro será necesario que personas ajenas a este no puedan acceder, para esto aquellos usuarios que tengan privilegios de acceso previamente definido por la empresa deberán ser confirmados ya que solo estos tendrán accesibilidad a aquellos módulos donde se hayan asignado dichos permisos y estos tengan vigencia.
- **Eficiencia y fiabilidad.** Se diseñará un sistema que no use excesivos recursos ya sea en comprar licencias de programas y que este presente el mínimo de fallas.

### **3.2.2. Costo**

Los costos fueron estimados en la viabilidad económica en este caso el único costo que sería tomado en cuenta sería la adquisición de la licencia del software SQLServer 2008.

## **CAPÍTULO IV**

### **4. ESPECIFICACIONES Y HERRAMIENTAS PARA LA SOLUCIÓN**

#### **4.1. Especificaciones Técnicas**

Se especifican los requerimientos mínimos que se deben tener para que la aplicación funcione correctamente tanto en hardware como en software.

##### **4.1.1. Requerimientos en Hardware**

Los requerimientos mínimos y las características recomendadas para servidores y clientes se detallan en la siguiente Tabla 4.1

N.	Dispositivo	Características	
		Mínimas	Recomendadas
1	Memoria	2 GB de RAM	4 GB de RAM
2	Disco duro	160 GB	500 GB
3	Procesador	Dual Core 2.9GHz.	Core i3.
4	Pantalla	800 x 600 píxel	1280 x 1024 píxel
5	Ancho de Banda	512Kbps	1024Kbps

**Tabla 4.1** Requerimientos por grado de dificultad.

#### 4.1.2. Requerimientos en Software

Los requerimientos basados en software se detallan en la siguiente Tabla 4.2.

No	Dispositivo	Características
1	Base de Datos	SQLSever 2008
2	Plataforma	Windows
3	Distribución	Server 2003 o Superiores
4	Servidor Web	IIS 7.5 o superiores
5	Framework	Microsoft .Net Framework 3.5 o superiores

**Tabla 4.2** Componentes basados en Software.

## **4.2. Herramientas basadas en Software**

### **4.2.1. Ambiente de desarrollo (IDEs)**

Para la implementación de las clases, funciones y métodos se utilizó el IDE Microsoft Visual Studio 2008 Versión 9.0.30729.1 SP1 en C# como lenguaje de programación, además de ciertos componentes y tecnologías orientadas al desarrollo web.

### **4.2.2. Librerías y componentes**

#### **CRYSTAL REPORTS BASIC**

Componente de Visual Studio 2008 que viene incluida dentro del mismo IDE. Crystal Report es un producto para la creación e integración de reportes con datos provenientes de múltiples fuente de datos tal como se explica en [6]. Para mas información revise la siguiente referencia [7].

#### **SQL SERVER 2008**

Motor de Base de datos utilizado para la elaboración y mantenimiento de datos. Para más información revise [8].

#### **AJAX y JQUERY**

jQuery es una librería de javascript que ayuda a simplificar la manera de interactuar los documentos html [9]. Ajax por su parte es una técnica de desarrollo web para crear aplicaciones interactivas [10].

El uso de JQuery como framework de desarrollo facilita el uso de AJAX integrado como parte de su solución.

### **AJAX CONTROL TOOLKIT**

AJAX Control Toolkit es un control o componente desarrollado para ASP cuya aplicación es el uso de controles que faciliten la creación de páginas interactivas. Para más detalles ver la siguiente referencia [11].



## **CAPÍTULO IV**

### **5. DISEÑO Y METODOLOGIA UTILIZADA**

#### **5.1. Aplicando arquitectura MVC**

El modelo está instalado en un servidor que se encuentra dentro de la oficina de la Unidad de Postgrado, y desde allí se brinda soporte al sistema local y web. Esta arquitectura consiste en que un cliente realiza peticiones al servidor y este le da las repuestas según la petición enviada, las repuestas del servidor por lo general son serializadas en formato json donde estos son tomados de Procedimientos almacenados que se encuentran en la base de datos MySQL.

#### **5.2. Modelo lógico de la base de datos**

El sistema de Administración de Unidades de Postgrado cuenta con un conjunto de tablas que permite almacenar datos por separado y

relacionar los datos de las diferentes entidades como persona, tesis, tutoría entre otros.

En la figura 5.1 se muestra la relación lógica entre las tablas y en el “anexo A” el ddl de la misma.

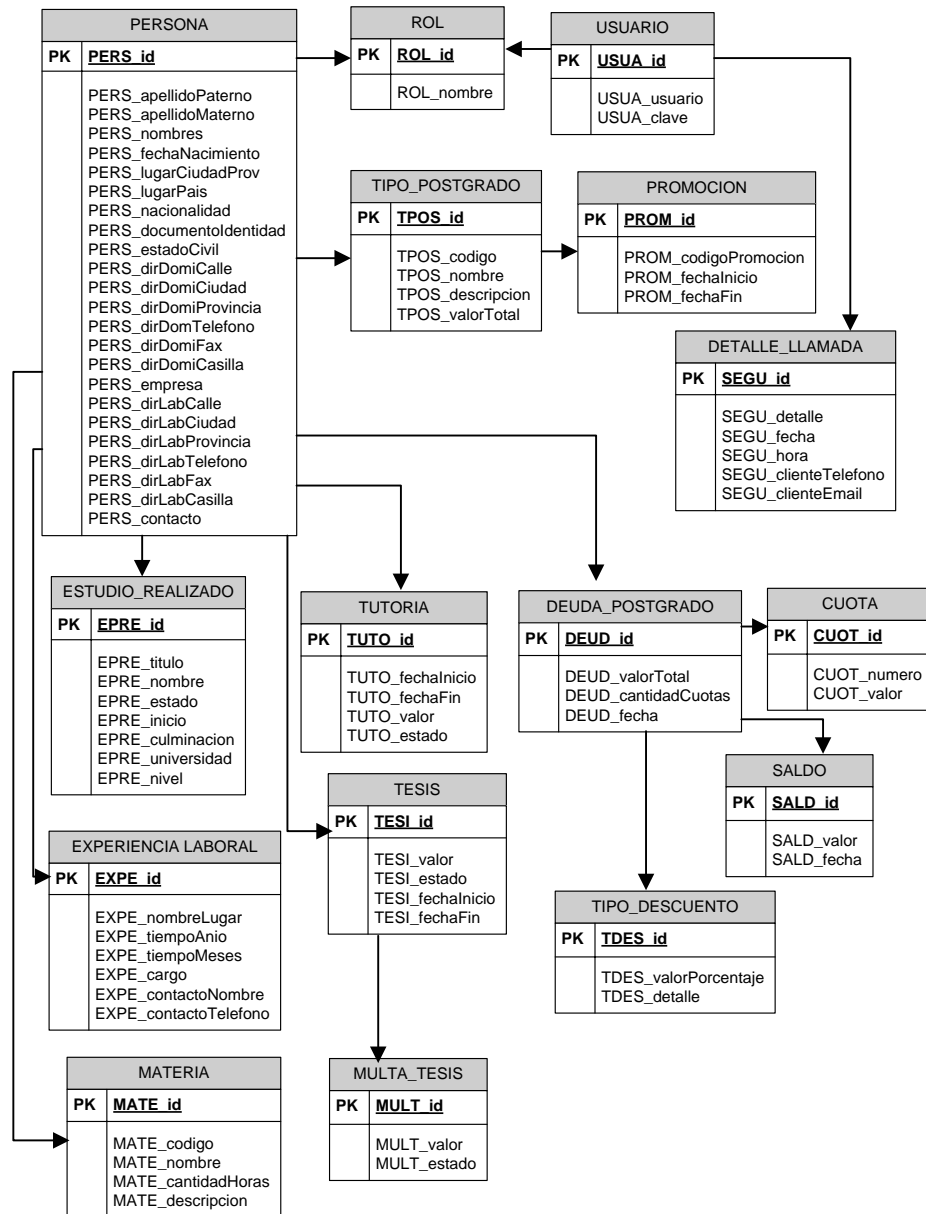


Figura 5.1: Modelo lógico de la base de datos.

A continuación se describen las tablas más relevantes:

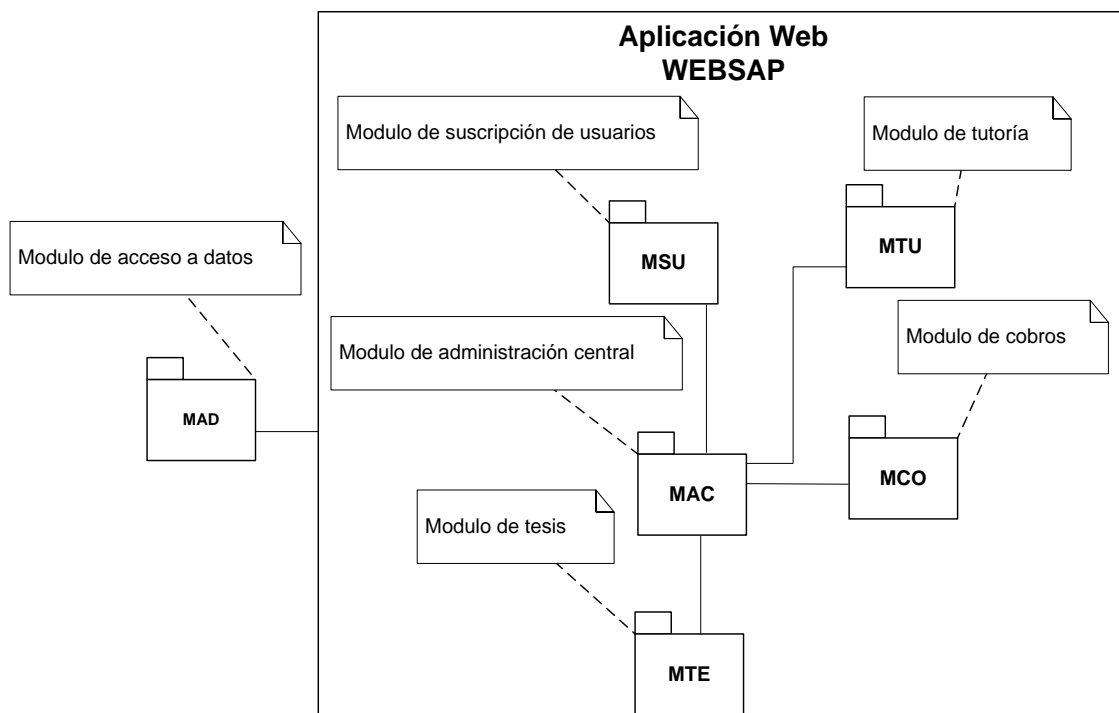
- **PERSONA:** Almacena la información sobre los datos personales de las personas que actúan en el sistema, como son estudiante, docente, administrador y secretarias.
- **POSTGRADO:** En esta tabla se almacena la información de los postgrados que se ofrecen en la Unidad.
- **PROMOCION:** En esta tabla es para identificar mediante un número, el cual representa que un grupo de estudiantes compartieron estudios durante un periodo determinado. Cada tipo de Postgrado tiene una promoción.
- **DETALLE\_LLAMADA:** En esta tabla se almacena la información que guarda un usuario del sistema (Ej. Secretaria) cuando solicita a la persona que llama a preguntar información sobre los Postgrados que ofrecen.
- **EXPERIENCIA\_LABORAL:** En esta tabla se almacenan todos los datos necesarios de un estudiante o docente si tienen un trabajo actual y sus antiguos trabajos
- **ESTUDIOS\_REALIZADOS:** En esta tabla se almacena la información sobre las instituciones donde el usuario a realizado sus estudios, así como el periodo de estudio.

- **TESIS:** En esta tabla se detalla el periodo que un estudiante tiene activo su proceso de tesis.
- **TUTORIA:** En esta tabla se detalla el periodo que un estudiante tiene activo su proceso de tutoria.
- **DEUDA\_POSTGRADO:** En esta tabla se detalla el valor que tiene el Postgrado seleccionado por el estudiante.
- **TIPO\_DESCUENTO:** En esta tabla se detalla el valor de descuento que un estudiante puede obtener.

### **5.3. Módulos del Sistema**

Los módulos diseñados e implementados en el Sistema de Administración de Unidades de Postgrado:

- Módulo de Administración Central (MAC).
- Modulo de Suscripción de Usuarios (MSU).
- Módulo de Tutoría (MTU).
- Módulo de Tesis (MTE).
- Módulo de Cobros (MCO).



**Figura 5.2** Módulos del Sistema.

### **MÓDULO DE ADMINISTRACIÓN CENTRAL (MAC)**

Se utiliza para manejar los diferentes roles que puede tener el usuario en la aplicación, y por ende las operaciones que este usuario pueda realizar dependiendo del rol o roles que posea.

### **MÓDULO DE SUSCRIPCIÓN DE USUARIOS (MSU)**

Es un módulo para la administración de Usuarios, además maneja la suscripción de los usuarios visitantes a la Aplicación Web.

### **MÓDULO DE TESIS (MTE)**

Se lleva el control del proceso de tesis que se maneja en el Centro de Postgrados, definición de estados y control de fechas del proceso.

### **MÓDULO DE TUTORIAS (MTU)**

En este módulo es para la presentación de estados del proceso de tutorías, control de fechas del servicio.

### **MÓDULO DE TUTORIAS (MCO)**

En este módulo es para la presentación de estados de los pagos que realizan los estudiantes por el servicio de postgrados.

### **MÓDULO DE ACCESO A DATOS (MCO)**

En este módulo es para la presentación de la información de la base de datos del sistema.

#### **5.3.1. Definición de roles en los módulos**

En la siguiente tabla (Tabla 5.1), se muestran las funciones que ejerce cada usuario del Centro de Postgrados.

Cada usuario registrado en el Sistema, tiene la posibilidad de tener de uno o más perfiles, de tal manera que pueden ejercer varios roles, dependiendo de las políticas especificadas por el Centro de Postgrados.

Los usuarios son los siguientes:

- Director del Centro de Postgrados
- Asistente del Centro de Postgrado
- Docente
- Estudiante

En la siguiente table se describe cada uno de los usuarios:

	<b>Función</b>	
Usuario	Dentro del negocio	Dentro del Sistema
Director del Centro de Postgrados	Representante del Centro de Postgrados encargado de planificar y elaborar las normas para los demás miembros, con la finalidad de llevar mejor organización.	Administrador General
Asistente del Centro de Postgrado	Ayudante del Director del Centro de postgrados que sigue las normas planificadas para la organización exitosa de los servicios que ofrecen.	Secretaria
Docente	Usuario que esta vinculado en los servicios que ofrece el Centro de postgrados	Docente
Estudiante	Usuario que se estudia un Postgrado.	Conferencista

**Tabla 5.1.** Definición de Roles en los Módulos

## 5.4. Casos de Estudio

### 5.4.1. Especificación de Actores

<b>Nombre:</b>	Administrador
<b>Descripción:</b>	Es aquel que estará a cargo de la administración del sistema.
<b>Notas:</b>	Es definido al momento de configurar el sistema y a través de la administración de nuevos usuarios.

<b>Nombre:</b>	Secretaria
<b>Descripción:</b>	Es aquel que estará a cargo del ingreso de los nuevos usuarios, sean estos administradores, secretarias, estudiantes, profesores y del ingreso de todos los datos referentes al sistema.
<b>Notas:</b>	Es definido por el administrador del sitio.

<b>Nombre:</b>	Estudiante
<b>Descripción:</b>	Es aquel que se registra en una de las maestrías o diplomado que ofrece la institución.
<b>Notas:</b>	Es definido por la secretaria al momento de registra una nueva persona.

<b>Nombre:</b>	Docente
<b>Descripción:</b>	Es aquel que dictara una materia de la maestría o diplomado que ofrece la institución.
<b>Notas:</b>	Es definido por la secretaria al momento de ingresar una nueva persona.



A continuación se define una lista de casos de uso que han sido considerados para la resolución de este proyecto.

1. Registro de administrador.
2. Registro de secretaria.
3. Registro de estudiante.
4. Registro de docente.
5. Registro de tutor.
6. Registro de deudas.
7. Registro de tesis.
8. Consulta de administrador.
9. Consulta de secretaria.
10. Consulta de docente.
11. Consulta de tutor.
12. Consulta de deudas.
13. Consulta de tesis.
14. Detalles de administrador.
15. Detalles de secretaria.
16. Detalles de estudiante.
17. Detalles de docente.

Para cada uno de estos casos se describe la funcionalidad requerida para lograr los objetivos de este proyecto.

<b>Nombre</b>	1. Registro de administrador
<b>Detalle</b>	Se registrará a una persona como administrador para que tenga acceso a todo el sistema.
<b>Notas</b>	Se deberá llenar todos los campos correspondientes para el registro de un nuevo administrador.
<b>Valor medible</b>	Se registró un Administrador o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente un administrador en la base de datos.</li> <li>• Ya existe un administrador con el mismo usuario.</li> <li>• No se registró correctamente el administrador en la base de datos.</li> </ul>

<b>Nombre</b>	2. Registro de secretaria
<b>Detalle</b>	Se registrará a una persona como secretaria para que tenga acceso a ciertas partes del sistema, como es el registro de estudiantes, nuevas usuarios y docentes, como la administración de las postgrados.
<b>Notas</b>	Se deberá llenar todos los campos correspondientes para el registro de una nueva secretaria.
<b>Valor medible</b>	Se registró una Secretaria o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente una secretaria en la base de datos.</li> <li>• Ya existe una secretaria con el mismo usuario.</li> <li>• No se registró correctamente la secretaria en la base de datos.</li> </ul>

<b>Nombre</b>	3. Registro de estudiante
<b>Detalle</b>	Se registrará a una persona como estudiante para que tenga acceso a las partes académicas del sistema el sistema.
<b>Notas</b>	Se deberá llenar todos los campos correspondientes para el registro de un nuevo estudiante.
<b>Valor medible</b>	Se registró un Estudiante o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente un estudiante en la base de datos.</li> <li>• Ya existe un estudiante con el mismo usuario.</li> <li>• No se registró correctamente el Estudiante en la base de datos.</li> </ul>

<b>Nombre</b>	4. Registro de docente
<b>Detalle</b>	Se registrará a una persona como docente para que tenga acceso a la parte académica para el docente en el sistema.
<b>Notas</b>	Se deberá llenar todos los campos correspondientes para el registro de un nuevo docente.
<b>Valor medible</b>	Se registró un Docente o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente un docente en la base de datos.</li> <li>• Ya existe un docente con el mismo usuario.</li> <li>• No se registró correctamente el Docente en la base de datos.</li> </ul>

<b>Nombre</b>	5. Registro de tutoría
<b>Detalle</b>	Se registrará a un docente como tutor para que de asesoramiento a un estudiante en el desarrollo de su tesis.
<b>Notas</b>	Se deberá llenar los campos en el cual el docente puede dar una tutoría.
<b>Valor medible</b>	Se registró una tutoría o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente una tutoría en la base de datos.</li> <li>• Ya existe una tutoría con el mismo docente.</li> <li>• Ya existe una tutoría con el mismo estudiante.</li> <li>• No se registro correctamente la tutoría en la base de datos.</li> </ul>

<b>Nombre</b>	6. Registro de deuda
<b>Detalle</b>	Se registrará una deuda al estudiante que se registre en una maestría distada por la institución.
<b>Notas</b>	Se deberá llenar los campos correspondientes a la deuda del estudiante.
<b>Valor medible</b>	Se registró una deuda o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente una deuda en la base de datos.</li> <li>• Ya existe una deuda de la maestría.</li> <li>• Ya existe una deuda de tesis.</li> <li>• Ya existe una deuda de tutoría.</li> <li>• No se registro correctamente la deuda en la base de datos.</li> </ul>

<b>Nombre</b>	7. Registro de tesis
<b>Detalle</b>	Se registrará una tesis al estudiante que ya cumpla con los requisitos para su elaboración.
<b>Notas</b>	Se deberá llenar los campos correspondientes para el registro de una tesis. Se deberá confirmar que el estudiante cumpla con todos los requisitos para el registro de la tesis.
<b>Valor medible</b>	Se registró una tesis o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se registró correctamente una tesis en la base de datos.</li> <li>• Ya existe una tesis para el estudiante.</li> <li>• No se registro correctamente la tesis en la base de datos.</li> </ul>

<b>Nombre</b>	8. Consulta de administrador
<b>Detalle</b>	Se consultará si el administrador existe en la base de datos, si existe se mostrará su información.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula.
<b>Valor medible</b>	El administrador existe o no en la base de datos.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• La información del administrador fue mostrada correctamente.</li> <li>• No existe el administrador en la base de datos.</li> <li>• No se puede mostrar la información del administrador.</li> </ul>

<b>Nombre</b>	9. Consulta de secretaria
<b>Detalle</b>	Se consultará si la secretaria existe en la base de datos, si existe se mostrará su información.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula.
<b>Valor medible</b>	La secretaria existe o no en la base de datos.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• La información de la secretaria fue mostrada correctamente.</li> <li>• No existe la secretaria en la base de datos.</li> <li>• No se puede mostrar la información de la secretaria.</li> </ul>

<b>Nombre</b>	10. Consulta de docente
<b>Detalle</b>	Se consultará si el docente existe en la base de datos, si existe se mostrará su información.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula.
<b>Valor medible</b>	El docente existe o no en la base de datos.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• La información del docente fue mostrada correctamente.</li> <li>• No existe el docente en la base de datos.</li> <li>• No se puede mostrar la información del docente.</li> </ul>

<b>Nombre</b>	11. Consulta de tutor
<b>Detalle</b>	Se consultará si un tutor fue asignado a un estudiante, si existe se mostrará su información.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula del estudiante.
<b>Valor medible</b>	El tutor esta o no asignado a un estudiante.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• La información del tutor fue mostrada correctamente.</li> <li>• No existe tutor para el estudiante.</li> <li>• No se puede mostrar la información del tutor.</li> </ul>

<b>Nombre</b>	12. Consulta de deudas
<b>Detalle</b>	Se consultará si el estudiante tiene deuda con la institución, si existe se mostrará su información.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula del estudiante.
<b>Valor medible</b>	El estudiante tiene o no una deuda con la institución.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• La información de la duda fue mostrada correctamente.</li> <li>• No existe deuda para el estudiante.</li> <li>• No se puede mostrar la información de la deuda.</li> </ul>

<b>Nombre</b>	13. Consulta de tesis
<b>Detalle</b>	Se consultará si el estudiante tiene una tesis asignada, si existe se mostrará su información.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula del estudiante.
<b>Valor medible</b>	El estudiante tiene o no asignada una tesis.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• La información de la tesis fue mostrada correctamente.</li> <li>• No existe tesis para el estudiante.</li> <li>• No se puede mostrar la información de la tesis.</li> </ul>

<b>Nombre</b>	14. Detalles de administrador
<b>Detalle</b>	Se debe haber hecho consulta de administrador.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula se podrá mostrar los detalles del mismo.
<b>Valor medible</b>	Se mostrará los detalles del administrador o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se presentó toda la información del administrador.</li> <li>• No existe información del administrador.</li> </ul>



<b>Nombre</b>	15. Detalles de secretaria
<b>Detalle</b>	Se debe haber hecho consulta de secretaria.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula se podrá mostrar los detalles del mismo.
<b>Valor medible</b>	Se mostrará los detalles de la secretaria o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se presentó toda la información de la secretaria.</li> <li>• No existe información de la secretaria.</li> </ul>

<b>Nombre</b>	16. Detalles de estudiante.
<b>Detalle</b>	Se debe haber hecho consulta de estudiante.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula se podrá mostrar los detalles del mismo.
<b>Valor medible</b>	Se mostrará los detalles del estudiante o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se presentó toda la información del estudiante.</li> <li>• No existe información del estudiante.</li> </ul>

<b>Nombre</b>	17. Detalles de docente.
<b>Detalle</b>	Se debe haber hecho consulta de docente.
<b>Notas</b>	Se consultará por medio del nombre, apellido o cedula se podrá mostrar los detalles del mismo.
<b>Valor medible</b>	Se mostrará los detalles del docente o no.
<b>Escenarios</b>	<ul style="list-style-type: none"> <li>• Se presentó toda la información del docente.</li> <li>• No existe información del docente.</li> </ul>

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

1. Este sistema evitara la acumulación de carpetas en las oficinas de la institución, ya que no existía un software que les ayudara a realizar su trabajo diario y corriendo el peligro que se pierdan estos documentos por estar a la intemperie y por la manipulación de ellos.
2. Ayudara a que los estudiantes no se aglomeren en las oficinas para pedir sus calificaciones, ya que las podrán consultar en línea.
3. Con este sistema se podrá llevar un mejor control de las operaciones financieras que tiene la institución y un mejor control de las asignaciones de los tutores a los estudiantes que lo requieran para poder terminar con éxito sus tesis de la maestría.

## Recomendaciones

1. Es importante por el lado de un programador siempre tener un respaldo de su código ya que puede haber al tipo de problema con los equipos en que se trabaja y se perderá todo lo ya trabajado.
2. Hay que tener un equipo dedicado al sistema, para no tener que instalarlo en una PC en la que estén trabajando a diario.
3. Ofrecer personalización en la visualización gráfica de las pantallas, de tal manera que permita un control por parte del usuario en el tipo de gráficos y paleta de colores a mostrar con la finalidad de que puedan obtener una mejor apreciación.

# APENDICE A:

## DISEÑO DE LA BASE DE DATOS (DDL)

```
USE [Sistema_Postgrados]
GO
/***** Object: Table [dbo].[Persona]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Persona](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [ApellidoPaterno] [nvarchar](50) NULL,
    [ApellidoMaterno] [nvarchar](50) NULL,
    [Nombres] [nvarchar](50) NULL,
    [FechaNacimiento] [datetime] NULL,
    [LugarCiudadProv] [nvarchar](50) NULL,
    [LugarPais] [nvarchar](50) NULL,
    [Nacionalidad] [nvarchar](50) NULL,
    [DocumentoIdentidad] [nvarchar](50) NULL,
    [EstadoCivil] [nvarchar](50) NULL,
    [DirDomiCalle] [nvarchar](50) NULL,
    [DirDomiCiudad] [nvarchar](50) NULL,
    [DirDomiProvincia] [nvarchar](50) NULL,
    [DirDomTelefono] [nvarchar](50) NULL,
    [DirDomiFax] [nvarchar](50) NULL,
    [DirDomiCasilla] [nvarchar](50) NULL,
    [Empresa] [nvarchar](50) NULL,
    [DirLabCalle] [nvarchar](50) NULL,
    [DirLabCidudad] [nvarchar](50) NULL,
    [DirLabProvincia] [nvarchar](50) NULL,
    [DirLabTelefono] [nvarchar](50) NULL,
    [DirLabFax] [nvarchar](50) NULL,
    [DirLabCasilla] [nvarchar](50) NULL,
    [Contacto] [nvarchar](50) NULL,
    [InglesPorcentajeHabla] [int] NULL,
    [InglesPorcentajeEscribe] [int] NULL,
    [EspanolPorcentajeEscribe] [int] NULL,
    [EspanolPorcentajeHabla] [int] NULL,
    [FrancesPorcentajeHable] [int] NULL,
    [FrancesPorcentajeEscribe] [int] NULL,
    [OtroPorcentajeHabla] [int] NULL,
    [OtroPorcentajeEscribe] [int] NULL,
    [OtroIdioma] [nvarchar](100) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
    CONSTRAINT [PK_Persona] PRIMARY KEY CLUSTERED
```

```

(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[TipoPostgrado]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TipoPostgrado](
    [Id] [int] NOT NULL,
    [Codigo] [nvarchar](50) NOT NULL,
    [Nombre] [nvarchar](50) NULL,
    [Descripcion] [nvarchar](50) NULL,
    [ValorTotal] [float] NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Tipo_Postgrado] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Rol]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Rol](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Rol] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Promocion]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Promocion](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdTipoPostgrado] [int] NOT NULL,
    [CodigoPromocion] [nvarchar](50) NOT NULL,
    [FechaInicio] [datetime] NOT NULL,

```

```

[FechaFin] [datetime] NOT NULL,
[ValorCuotalInicial] [decimal](18, 2) NOT NULL,
[ValorTotal] [decimal](18, 2) NOT NULL,
[EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Promocion] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ExperienciaLaboral] Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ExperienciaLaboral](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdPersona] [int] NULL,
    [NombreLugar] [nvarchar](50) NULL,
    [TiempoAnio] [int] NULL,
    [TiempoMeses] [int] NULL,
    [Cargo] [nvarchar](50) NULL,
    [ContactoNombre] [nvarchar](50) NULL,
    [TelefonoContacto] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Experiencia_Laboral] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[EstudioRealizado] Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[EstudioRealizado](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdPersona] [int] NULL,
    [Titulo] [nvarchar](10) NULL,
    [Nombre] [nvarchar](50) NULL,
    [Estado] [nvarchar](50) NULL,
    [Inicio] [datetime] NULL,
    [Culminacion] [datetime] NULL,
    [Universidad] [nvarchar](50) NULL,
    [Nivel] [nvarchar](max) NOT NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Estudio_Pregado] PRIMARY KEY CLUSTERED
(
    [Id] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Estudiante]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Estudiante](
    [Id] [int] NOT NULL,
    [IdPersona] [int] NOT NULL,
    [Matricula] [varchar](50) NULL,
    [EstadoRegistro] [smallint] NULL,
    CONSTRAINT [PK_Persona_Estudiante] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Docente]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Docente](
    [Id] [int] NOT NULL,
    [IdPersona] [int] NOT NULL,
    [EstadoRegistro] [smallint] NOT NULL,
    CONSTRAINT [PK_Persona_Docente] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Usuario]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Usuario](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdPersona] [int] NOT NULL,
    [IdRol] [int] NOT NULL,
    [Usuario] [nvarchar](50) NULL,
    [Clave] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,

```

```

CONSTRAINT [PK_Usuario] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Tutoria]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tutoria](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdDocente] [int] NOT NULL,
    [IdEstudiante] [int] NOT NULL,
    [FechaInicio] [datetime] NULL,
    [FechaFin] [datetime] NULL,
    [Valor] [float] NULL,
    [Estado] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Tutoria] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Deuda_Postgrado]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Deuda_Postgrado](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdEstudiante] [int] NOT NULL,
    [Fecha] [nvarchar](max) NOT NULL,
    [ValorTotal] [float] NULL,
    [CantidadCuotas] [int] NULL,
    [Saldo] [nvarchar](max) NOT NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Deuda_Postgrado] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Detalle_Llamada]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```



```

CREATE TABLE [dbo].[Detalle_Llamada](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdUsuario] [int] NOT NULL,
    [Detalle] [nvarchar](50) NULL,
    [Fecha] [datetime] NULL,
    [Hora] [datetime] NULL,
    [TelefonoCliente] [nvarchar](50) NULL,
    [EmailCliente] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Detalle_Llamada] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Materia]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Materia](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdDocente] [int] NOT NULL,
    [Codigo] [nvarchar](50) NULL,
    [Nombre] [nvarchar](50) NULL,
    [CantidadHoras] [int] NULL,
    [Descripcion] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Materia] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[EstudiantePromocion]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[EstudiantePromocion](
    [Id] [int] NOT NULL,
    [IdEstudiante] [int] NOT NULL,
    [IdPromocion] [int] NOT NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_EstudiantePromocion_1] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

/***** Object: Table [dbo].[Tesis]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tesis](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdEstudiante] [int] NOT NULL,
    [Valor] [float] NULL,
    [FechaInicio] [datetime] NULL,
    [FechaFin] [datetime] NULL,
    [Docente] [int] NULL,
    [Estado] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
    CONSTRAINT [PK_Tesis] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TipoDescuento]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TipoDescuento](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdDeuda] [int] NOT NULL,
    [IdEstudiante] [int] NOT NULL,
    [ValorPorcentaje] [int] NULL,
    [Detalle] [nvarchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,
    CONSTRAINT [PK_Tipo_Descuento] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[MultiTesis]  Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[MultiTesis](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdTesis] [int] NOT NULL,
    [Valor] [float] NULL,
    [Estado] [varchar](50) NULL,
    [EstadoRegistro] [smallint] NOT NULL,

```

```

CONSTRAINT [PK_Multa_Tesis] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Cuota]    Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Cuota](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdDeuda] [int] NOT NULL,
    [Numero] [int] NULL,
    [Valor] [float] NULL,
    [Fecha] [date] NULL,
    [Tipo] [tinyint] NOT NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Cuota] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Calificacion]    Script Date: 02/17/2011 23:23:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Calificacion](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdEstudiante] [int] NULL,
    [IdMateria] [int] NOT NULL,
    [Valor] [int] NULL,
    [EstadoRegistro] [smallint] NOT NULL,
CONSTRAINT [PK_Nota_Calificacion] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: ForeignKey [FK_Calificacion_Estudiante]    Script Date: 02/17/2011 23:23:15
*****/
ALTER TABLE [dbo].[Calificacion] WITH CHECK ADD CONSTRAINT
[FK_Calificacion_Estudiante] FOREIGN KEY([IdEstudiante])
REFERENCES [dbo].[Estudiante] ([Id])
GO

```

```

ALTER TABLE [dbo].[Calificacion] CHECK CONSTRAINT [FK_Calificacion_Estudiante]
GO
/***** Object: ForeignKey [FK_MateriaNota_Calificacion]  Script Date: 02/17/2011 23:23:15
*****/
ALTER TABLE [dbo].[Calificacion] WITH CHECK ADD CONSTRAINT
[FK_MateriaNota_Calificacion] FOREIGN KEY([IdMateria])
REFERENCES [dbo].[Materia] ([Id])
GO
ALTER TABLE [dbo].[Calificacion] CHECK CONSTRAINT [FK_MateriaNota_Calificacion]
GO
/***** Object: ForeignKey [FK_Deuda_PostgradoCuota]  Script Date: 02/17/2011 23:23:15
*****/
ALTER TABLE [dbo].[Cuota] WITH CHECK ADD CONSTRAINT
[FK_Deuda_PostgradoCuota] FOREIGN KEY([IdDeuda])
REFERENCES [dbo].[Deuda_Postgrado] ([Id])
GO
ALTER TABLE [dbo].[Cuota] CHECK CONSTRAINT [FK_Deuda_PostgradoCuota]
GO
/***** Object: ForeignKey [FK_UsuarioDetalle_Llamada]  Script Date: 02/17/2011 23:23:15
*****/
ALTER TABLE [dbo].[Detalle_Llamada] WITH CHECK ADD CONSTRAINT
[FK_UsuarioDetalle_Llamada] FOREIGN KEY([IdUsuario])
REFERENCES [dbo].[Usuario] ([Id])
GO
ALTER TABLE [dbo].[Detalle_Llamada] CHECK CONSTRAINT
[FK_UsuarioDetalle_Llamada]
GO
/***** Object: ForeignKey [FK_EstudianteDeuda_Postgrado]  Script Date: 02/17/2011
23:23:15 *****/
ALTER TABLE [dbo].[Deuda_Postgrado] WITH CHECK ADD CONSTRAINT
[FK_EstudianteDeuda_Postgrado] FOREIGN KEY([IdEstudiante])
REFERENCES [dbo].[Estudiante] ([Id])
GO
ALTER TABLE [dbo].[Deuda_Postgrado] CHECK CONSTRAINT
[FK_EstudianteDeuda_Postgrado]
GO
/***** Object: ForeignKey [FK_Docente_Persona]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[Docente] WITH CHECK ADD CONSTRAINT [FK_Docente_Persona]
FOREIGN KEY([IdPersona])
REFERENCES [dbo].[Persona] ([Id])
GO
ALTER TABLE [dbo].[Docente] CHECK CONSTRAINT [FK_Docente_Persona]
GO
/***** Object: ForeignKey [FK_Estudiante_Persona]  Script Date: 02/17/2011 23:23:15
*****/
ALTER TABLE [dbo].[Estudiante] WITH CHECK ADD CONSTRAINT
[FK_Estudiante_Persona] FOREIGN KEY([IdPersona])
REFERENCES [dbo].[Persona] ([Id])
GO
ALTER TABLE [dbo].[Estudiante] CHECK CONSTRAINT [FK_Estudiante_Persona]
GO
/***** Object: ForeignKey [FK_EstudiantePromocion_Estudiante]  Script Date: 02/17/2011
23:23:15 *****/

```

```

ALTER TABLE [dbo].[EstudiantePromocion] WITH CHECK ADD CONSTRAINT
[FK_EstudiantePromocion_Estudiante] FOREIGN KEY([IdEstudiante])
REFERENCES [dbo].[Estudiante] ([Id])
GO
ALTER TABLE [dbo].[EstudiantePromocion] CHECK CONSTRAINT
[FK_EstudiantePromocion_Estudiante]
GO
/***** Object: ForeignKey [FK_EstudiantePromocion_Promocion]  Script Date: 02/17/2011
23:23:15 *****/
ALTER TABLE [dbo].[EstudiantePromocion] WITH CHECK ADD CONSTRAINT
[FK_EstudiantePromocion_Promocion] FOREIGN KEY([IdPromocion])
REFERENCES [dbo].[Promocion] ([Id])
GO
ALTER TABLE [dbo].[EstudiantePromocion] CHECK CONSTRAINT
[FK_EstudiantePromocion_Promocion]
GO
/***** Object: ForeignKey [FK_EstudioPregrado_Persona]  Script Date: 02/17/2011
23:23:15 *****/
ALTER TABLE [dbo].[EstudioRealizado] WITH CHECK ADD CONSTRAINT
[FK_EstudioPregrado_Persona] FOREIGN KEY([IdPersona])
REFERENCES [dbo].[Persona] ([Id])
GO
ALTER TABLE [dbo].[EstudioRealizado] CHECK CONSTRAINT
[FK_EstudioPregrado_Persona]
GO
/***** Object: ForeignKey [FK_ExperienciaLaboral_Persona]  Script Date: 02/17/2011
23:23:15 *****/
ALTER TABLE [dbo].[ExperienciaLaboral] WITH CHECK ADD CONSTRAINT
[FK_ExperienciaLaboral_Persona] FOREIGN KEY([IdPersona])
REFERENCES [dbo].[Persona] ([Id])
GO
ALTER TABLE [dbo].[ExperienciaLaboral] CHECK CONSTRAINT
[FK_ExperienciaLaboral_Persona]
GO
/***** Object: ForeignKey [FK_DocenteMateria]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[Materia] WITH CHECK ADD CONSTRAINT [FK_DocenteMateria]
FOREIGN KEY([IdDocente])
REFERENCES [dbo].[Docente] ([Id])
GO
ALTER TABLE [dbo].[Materia] CHECK CONSTRAINT [FK_DocenteMateria]
GO
/***** Object: ForeignKey [FK_TesisMultia_Tesis]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[MultiaTesis] WITH CHECK ADD CONSTRAINT
[FK_TesisMultia_Tesis] FOREIGN KEY([IdTesis])
REFERENCES [dbo].[Tesis] ([Id])
GO
ALTER TABLE [dbo].[MultiaTesis] CHECK CONSTRAINT [FK_TesisMultia_Tesis]
GO
/***** Object: ForeignKey [FK_Tipo_PostgradoPromocion]  Script Date: 02/17/2011
23:23:15 *****/
ALTER TABLE [dbo].[Promocion] WITH CHECK ADD CONSTRAINT
[FK_Tipo_PostgradoPromocion] FOREIGN KEY([IdTipoPostgrado])
REFERENCES [dbo].[TipoPostgrado] ([Id])

```

```

GO
ALTER TABLE [dbo].[Promocion] CHECK CONSTRAINT [FK_Tipo_PostgradoPromocion]
GO
/***** Object: ForeignKey [FK_EstudianteTesis]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[Tesis] WITH CHECK ADD CONSTRAINT [FK_EstudianteTesis]
FOREIGN KEY([IdEstudiante])
REFERENCES [dbo].[Estudiante] ([Id])
GO
ALTER TABLE [dbo].[Tesis] CHECK CONSTRAINT [FK_EstudianteTesis]
GO
/***** Object: ForeignKey [FK_Deuda_PostgradoTipo_Descuento]  Script Date:
02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[TipoDescuento] WITH CHECK ADD CONSTRAINT
[FK_Deuda_PostgradoTipo_Descuento] FOREIGN KEY([IdDeuda])
REFERENCES [dbo].[Deuda_Postgrado] ([Id])
GO
ALTER TABLE [dbo].[TipoDescuento] CHECK CONSTRAINT
[FK_Deuda_PostgradoTipo_Descuento]
GO
/***** Object: ForeignKey [FK_EstudianteTipo_Descuento]  Script Date: 02/17/2011
23:23:15 *****/
ALTER TABLE [dbo].[TipoDescuento] WITH CHECK ADD CONSTRAINT
[FK_EstudianteTipo_Descuento] FOREIGN KEY([IdEstudiante])
REFERENCES [dbo].[Estudiante] ([Id])
GO
ALTER TABLE [dbo].[TipoDescuento] CHECK CONSTRAINT
[FK_EstudianteTipo_Descuento]
GO
/***** Object: ForeignKey [FK_DocenteTutoria]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[Tutoria] WITH CHECK ADD CONSTRAINT [FK_DocenteTutoria]
FOREIGN KEY([IdDocente])
REFERENCES [dbo].[Docente] ([Id])
GO
ALTER TABLE [dbo].[Tutoria] CHECK CONSTRAINT [FK_DocenteTutoria]
GO
/***** Object: ForeignKey [FK_EstudianteTutoria]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[Tutoria] WITH CHECK ADD CONSTRAINT [FK_EstudianteTutoria]
FOREIGN KEY([IdEstudiante])
REFERENCES [dbo].[Estudiante] ([Id])
GO
ALTER TABLE [dbo].[Tutoria] CHECK CONSTRAINT [FK_EstudianteTutoria]
GO
/***** Object: ForeignKey [FK_PersonaUsuario]  Script Date: 02/17/2011 23:23:15 *****/
ALTER TABLE [dbo].[Usuario] WITH CHECK ADD CONSTRAINT [FK_PersonaUsuario]
FOREIGN KEY([IdPersona])
REFERENCES [dbo].[Persona] ([Id])
GO
ALTER TABLE [dbo].[Usuario] CHECK CONSTRAINT [FK_PersonaUsuario]
GO

```

## APENDICE B:

### PRUEBAS DE USABILIDAD

#### HOJAS DE OBSERVACIÓN

<b>Usuario:</b>	<b>Administrador</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Ingreso de Usuarios	1	150	120	1
Modificación de Usuarios	2	120	90	1
Consulta de Usuarios	1	60	20	1
Consulta de Postgrados	1	30	10	2
Creación de Promociones	1	30	15	1
Ingreso de Pagos	2	120	90	1

<b>Usuario:</b>	<b>Asistente 1</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Ingreso de Usuarios	1	150	180	1
Modificación de Usuarios	2	120	100	1
Consulta de Usuarios	1	60	40	1
Consulta de Postgrados	1	30	20	2
Creación de Promociones	1	30	25	1
Ingreso de Pagos	2	120	100	1



<b>Usuario:</b>	<b>Asistente 2</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Ingreso de Usuarios	1	150	160	1
Modificación de Usuarios	2	120	110	1
Consulta de Usuarios	1	60	50	1
Consulta de Postgrados	1	30	30	2
Creación de Promociones	1	30	30	1
Ingreso de Pagos	1	120	120	1

Resultado de la hoja de observación.

<b>Funcionalidad</b>	<b>Tiempo Promedio (s)</b>
Ingreso de Usuarios	153.3
Modificación de Usuarios	100
Consulta de Usuarios	36.6
Consulta de Postgrados	20
Creación de Promociones	23.3
Ingreso de pagos	103.3

<b>Usuario:</b>	<b>Usuario 1</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	2
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

<b>Usuario:</b>	<b>Usuario 2</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

<b>Usuario:</b>	<b>Usuario 3</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

<b>Usuario:</b>	<b>Usuario 4</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

<b>Usuario:</b>	<b>Usuario 5</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	3	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

<b>Usuario:</b>	<b>Usuario 6</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	3	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	1

<b>Usuario:</b>	<b>Usuario 7</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	3	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	1

<b>Usuario:</b>	<b>Usuario 8</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	1
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

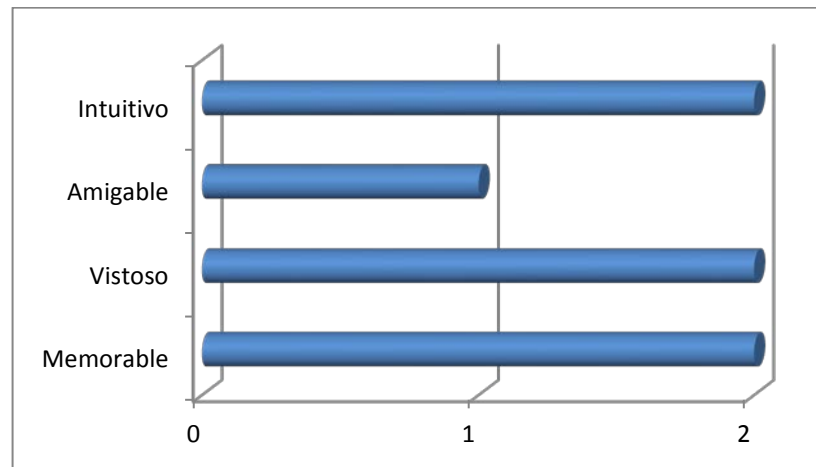
<b>Usuario:</b>	<b>Usuario 9</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	4
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

<b>Usuario:</b>	<b>Usuario 10</b>			
<b>Funcionalidad</b>	<b>Clics realizados</b>	<b>Tiempo (s)</b>		
		<b>Ingreso de datos (estimado)</b>	<b>Ingreso de datos</b>	<b>Ejecución del Sistema</b>
Descarga de Reportes de pagos	4	-	-	4
Consulta de pagos en línea	3	-	-	1
Consulta de últimas llamadas	1	-	-	2

# ENCUESTAS DE EVALUACIÓN A LOS USUARIOS

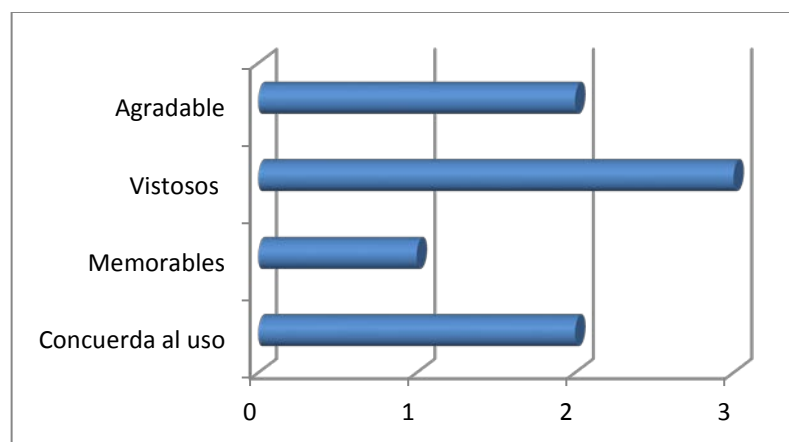
1) Los íconos de la barra de menú resultan:

Memorable	2
Vistoso	2
Amigable	1
Intuitivo	2



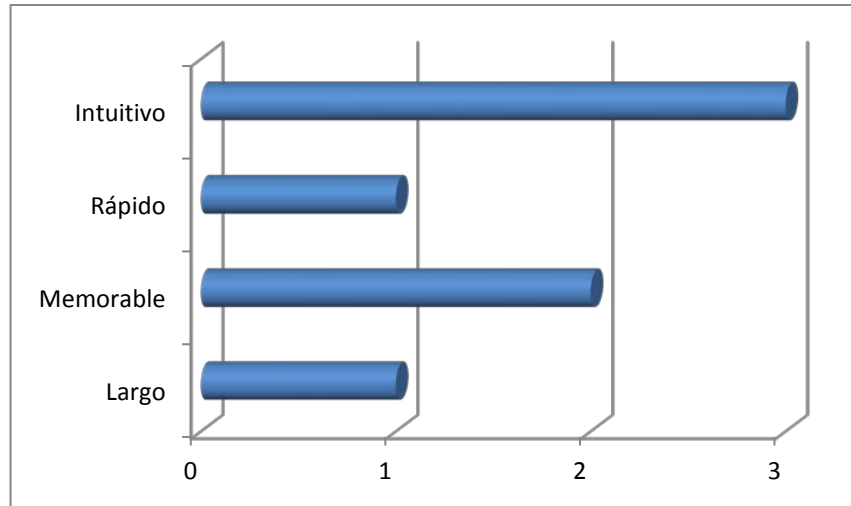
2) Los colores del sistema son:

Agradable	2
Vistosos	3
Memorables	1
Concuenda al uso	2



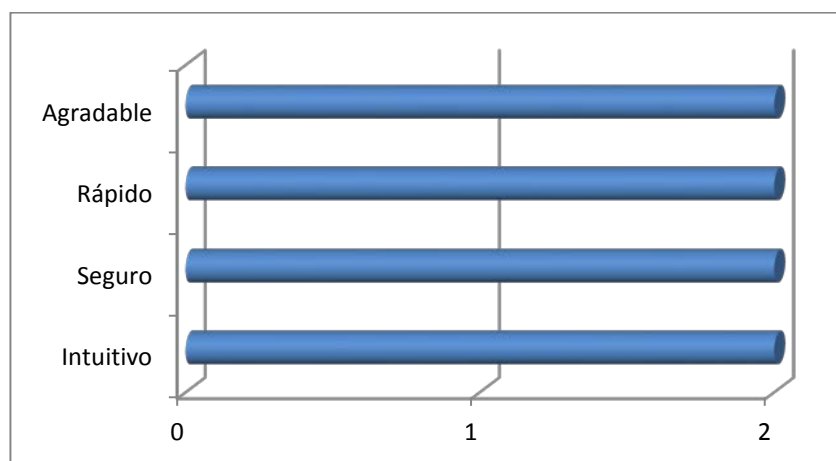
3) ¿Qué tal le pareció la ventana de ingreso de Usuarios?

Intuitivo	3
Rápido	1
Memorable	2
Largo	1



4) ¿Qué tal le pareció inicio de sesión del Sistema?

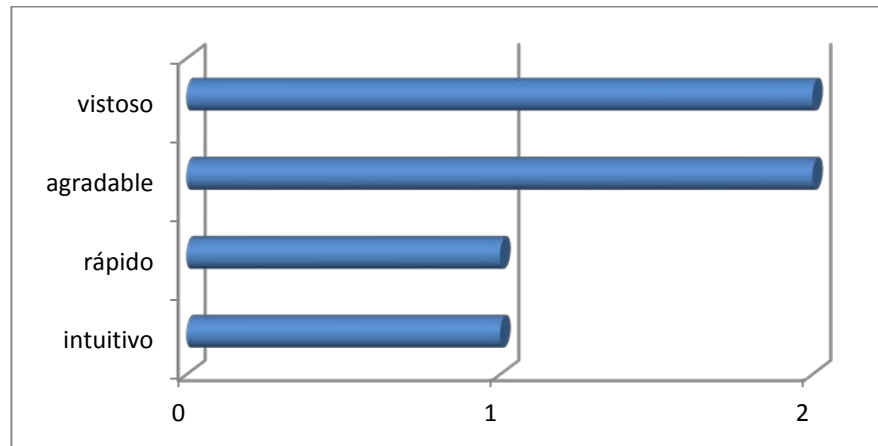
agradable	2
rápido	2
seguro	2
intuitivo	2





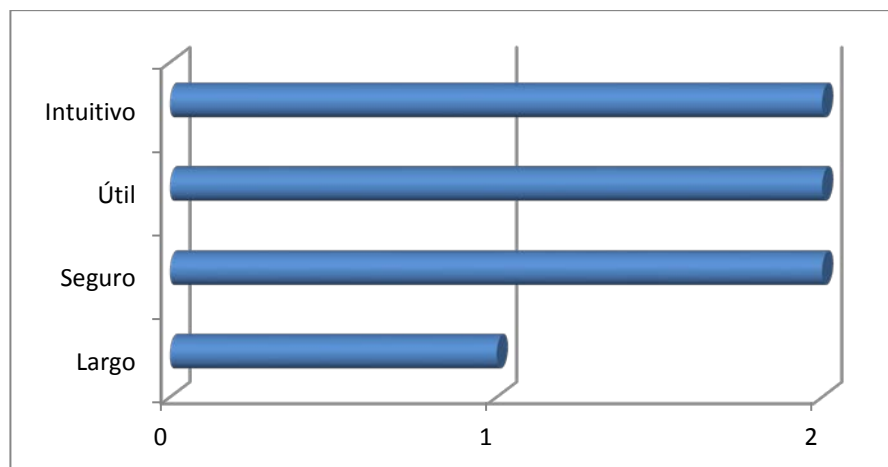
5) ¿Qué tal le pareció la interfaz de pagos?

Intuitivo	1
Rápido	1
Agradable	2
Vistoso	2



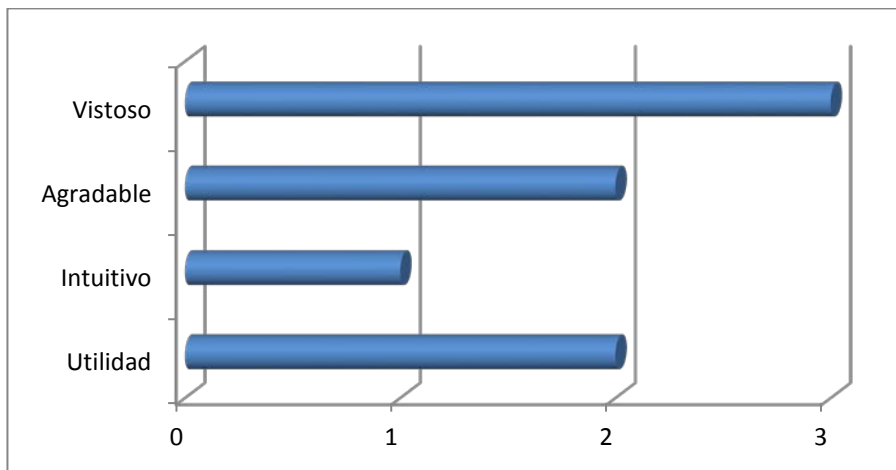
6) ¿Qué tal le pareció la interfaz de ingreso del detalle de llamadas?

Intuitivo	2
Útil	2
Seguro	2
Largo	1



7) ¿Qué tal le pareció la interfaz de consultas de pagos online?

Utilidad	2
Intuitivo	1
Agradable	2
Vistoso	3



# **GLOSARIO**

## **DDL**

El lenguaje de definición de datos (en inglés Data Definition Language, o DDL), es el que se encarga de la modificación de la estructura de los objetos de la base de datos.

## **ARQUITECTURA CLIENTE/SERVIDOR**

Consiste en un cliente que realiza peticiones a otro programa (el servidor) que le da respuestas.

## **FRAMEWORK**

Estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

## **SQLServer**

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario

## **QUERY**

Sentencia de búsqueda en una base de datos

## **STORE PROCEDURE o PROCEDIMIENTO ALMACENADO**

Es un programa el cual es almacenado físicamente en una base de datos.

## **IDE**

IDE como se conoce por sus siglas entorno de desarrollo integrado. Es un software que provee facilidades para desarrollar un programa ya sea este en cualquier lenguaje de programación.

## **JAVASCRIPT**

Lenguaje de tipo Scripting permite acceder a objetos de tipo html y manipularlos usando DOM (Document Object Model).

## **AJAX**

Asynchronous Javascript And Xml. Es una técnica de desarrollo web para crear aplicaciones interactivas.

## **JQUERY**

Es una biblioteca o framework de javascript que permite simplificar la manera de interactuar con javascript en documentos HTML.

## **C#**

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

## **.NET**

Es un framework de Microsoft que permite desarrollar todo tipo de aplicaciones de tipo web, desktop, móvil, etc.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Monografias, Publicaciones y temas de investigación. "Análisis y Diseño de Sistemas". Disponible en:  
<http://www.monografias.com/trabajos/anaydisis/anaydisis.shtml>  
Fecha visita: Noviembre 9 del 2010.
- [2] Diseño Web, "Definición de Diseño Web". Disponible en:  
[http://es.wikipedia.org/wiki/Dise%C3%B1o\\_web](http://es.wikipedia.org/wiki/Dise%C3%B1o_web)  
Fecha visita: Noviembre 9 del 2010.
- [3] Microsoft. Precios y Licencias. "Windows Compute Cluster Server 2003. Precios y ". Disponible en:  
<http://www.microsoft.com/spain/windowsserver2003/ccs/pricing.mspc>.  
Fecha visita: Noviembre 9 del 2010.
- [4] Microsoft, Precios y Licencias. "SQL Server 2008 - Esquemas de Licenciamiento". Disponible en:  
[http://download.microsoft.com/download/A/A/C/AACDC5CD-7A28-4493-BB0D-79FE286DF471/SQLServer2008\\_pricing.pdf](http://download.microsoft.com/download/A/A/C/AACDC5CD-7A28-4493-BB0D-79FE286DF471/SQLServer2008_pricing.pdf).  
Fecha visita: Noviembre 9 del 2010
- [5] Tipos de viabilidad. "Como evaluar un proyecto de inversion". Disponible en:  
<http://negociosyemprendimiento.com/como-evaluar-un-proyecto-de-inversion/>. Fecha visita: Noviembre 9 del 2010"
- [6] Crystal Reports, "Alta Productividad en creacion de reportes e integración de datos". Disponible en:

<http://www.crystalsolutions.com.ar/productos/crystalreports.html>.

Fecha visita: Noviembre 9 del 2010”

- [7] Crystal Reports, “Crystal Reports de VISUAL STUDIO 2008”. Disponible en:

[http://msdn.microsoft.com/es-es/library/bb126227\(v=VS.90\).aspx](http://msdn.microsoft.com/es-es/library/bb126227(v=VS.90).aspx).

Fecha visita: Noviembre 9 del 2010”

- [8] SQLServer 2008, “Build full-featured small-scale database solutions”

Disponible en:

<http://www.microsoft.com/sqlserver/2008/en/us/express.aspx>.

Fecha visita: Noviembre 10 del 2010.

- [9] JQuery, “Definición de JQuery” Disponible en:

<http://es.wikipedia.org/wiki/JQuery>.

Fecha visita: Noviembre 11 del 2010.

- [10] AJAX, “Definición de AJAX” Disponible en:

<http://es.wikipedia.org/wiki/AJAX>.

Fecha visita: Noviembre 11 del 2010.

- [11] AJAX Control Toolkit, “What is the ASP.NET AJAX Control Toolkit?”,

Disponible en:

<http://www.asp.net/ajax/ajaxcontroltoolkit/samples/>.

Fecha de visita: Noviembre 11 del 2010.

- [12] MVC, “Definición de MVC” Disponible en:

[http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

Fecha visita: Noviembre 17 del 2010.