



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE
PUEBLA**

**FACULTAD DE CIENCIAS
DE LA COMPUTACIÓN**

**Sistema de pre-inscripciones en web para
una Universidad Incorporada a la BUAP
(IESAC)**

T E S I N A

Para Obtener El Título De:

**INGENIERÍA EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A:

JOSE JULIAN RODRIGUEZ MEJORADA

ASESOR DE TESIS:

M.C. CARLOS ARMANDO RIOS ACEVEDO

2014



Agradecimientos

Quiero agradecer la dirección y el apoyo del M.C Carlos Armando Ríos Acevedo, sin el cual esta tesina no se hubiese emprendido.

A mis padres que sacrificaron mucho para hacer esto posible, por enseñarme de que siempre se pueden hacer la cosas, sólo es cuestión de querer hacerlas, por escucharme cuando necesitaba un consejo y acompañarme cuando lo necesitaba. A ellos más que a nadie.

A mis profesores, que me guiaron en los estudios y me formaron como profesional.

A mi esposa por apoyarme y ser perseverante todo el tiempo que duraron mis estudios y a mis hijas por ser mi principal motivo y razón de seguir adelante.



Índice General

1. Introducción	5
1.1. Importancia del proyecto.....	5
1.2. Objetivo.....	6
1.3. Objetivos Específicos	7
1.4. Justificación.....	7
1.5. Hipótesis.....	8
2. Marco Teórico.....	8
2.1. Conceptos Básicos.....	9
2.1.1. Base de Datos.....	9
2.1.2. Modelo Entidad-Relación	9
2.1.3. Entidad.....	10
2.1.4. Relación (interrelación).....	11
2.1.5. Atributo	12
2.1.6. Identificador	13
2.1.7. Metodología de diseño conceptual	13
2.1.8. Normalización.....	14
2.2. Sistema de Gestión de Bases de Datos (DBMS Database Management System)	17
2.2.1. Redundancia e inconsistencia de datos.	18
2.2.2. Dificultad para tener acceso a los datos.....	18
2.2.3. Aislamiento de los datos.....	18
2.2.4. Anomalías del acceso concurrente.....	18
2.2.5. Problemas de seguridad.....	19
2.2.6. Problemas de integridad.....	19
2.3. Acceso a Base de Datos.....	19
2.3.1. PDO (PHP Data Objects).....	19
2.4. Conceptos Básicos Web	20
2.4.1. Cliente - Servidor	20
2.4.2. Servidores Web.....	21
2.4.3. Apache.....	22



2.4.4.	Seguridad.....	23
2.4.5.	Hipertext Preprocesor (PHP).....	24
2.4.6.	JavaScript.....	25
2.4.7.	JQuery.....	26
3.	Metodología de Investigación.....	26
3.1.	Elementos de UML.....	30
	Diagrama de casos de uso.....	30
	Caso de uso.....	30
	Actor.....	31
	Descripción de casos de uso.....	31
	Diagrama de clases.....	32
	Clase.....	32
	Atributos.....	32
	Operaciones.....	33
	Asociaciones de clases.....	33
4.	Propuesta del Sistema.....	37
4.1.	Análisis y Diseño.....	37
4.1.1.	Requerimientos Funcionales.....	37
4.1.2.	Requerimientos No Funcionales.....	38
4.1.3.	Descripción General del Sistema.....	38
5.	Diseño del Sistema.....	55
5.1.	Diseño de la Base de Datos.....	55
5.1.1.	Diagrama Físico.....	55
5.1.2.	Diagrama Entidad – Relación.....	56
5.2.	Diseño de las Tablas de la Base de Datos.....	57
5.3.	Diseño de los Procedimientos Almacenados.....	62
5.4.	Diseño de Entradas.....	65
5.5.	Diseño de Salidas.....	75
6.	Conclusiones y Trabajos Futuros.....	78
6.1.	Conclusiones.....	78
6.2.	Trabajos Futuros.....	79
	Índice de Figuras.....	80
	Bibliografía.....	82



1. Introducción

1.1. Importancia del proyecto

Hoy en día, los sistemas informáticos juegan un papel muy importante en el desarrollo de las organizaciones y en el control de las operaciones administrativas y financieras entre otras, con la obtención de información inmediata y eficaz, la reducción de costos y tiempo en los diferentes procesos dentro de la estructura organizativa, determina el éxito y el alcance de los objetivos de cualquier empresa.

El presente proyecto consiste en el análisis, diseño e implementación de un sistema de pre-inscripción en línea, aplicable para la universidad del IESAC, sus dos modalidades, **estudiantes** y **administrativa**, que podrá ser consultado desde cualquier parte en un equipo con acceso a internet, a través del portal académico en la página del IESAC.

En sus dos modalidades, el sistema permitirá consultar y administrar los diferentes servicios que se mencionan a continuación:

1. Consulta del estado de inscripción (estudiante).
2. Administración y selección de horarios de materias del cuatrimestre.
3. Pre-inscripción del estudiante a un cuatrimestre.
4. Consulta y descarga de horario de materias del cuatrimestre.
5. Consulta de estudiantes pre-inscritos por cuatrimestre.

Es muy importante para el IESAC, apoyar a los estudiantes que estén al corriente en sus pagos y no tengan alguna restricción administrativa, evitándoles el traslado a las oficinas de la institución para la consulta de trámites administrativos, más aún con el antecedente de que el 40% de los estudiantes del IESAC, no son de la ciudad de Puebla.



La consulta fácil y rápida del estado administrativo en una interfaz amigable, permitirá al estudiante contar con el tiempo necesario para resolver su situación administrativa en tiempo y forma para que pueda pre-inscribirse.

De la misma manera la consulta del horario de materias del cuatrimestre, ayudará a los estudiantes para darse una idea global y concisa de la carga de materias del cuatrimestre a cursar.

El departamento escolar del IESAC, podrá generar reportes siempre actualizados, del número de estudiantes pre-inscritos en línea, para la toma de decisiones y principalmente la reducción de tiempos administrativos, ya que sólo se presentaran a las instalaciones los estudiantes con problemas administrativos que impidan su pre-inscripción por ejemplo: adeudo en colegiaturas, etc. También en la mejora tecnológica para la publicación y generación de los horarios de todas las carreras de la universidad por cuatrimestre.

La idea de contar con un sistema de pre-inscripción en línea, crea la necesidad de conocer las opiniones, dudas o comentarios de los estudiantes, por esta razón el sistema dispondrá de una sección de Contacto.

1.2. Objetivo

Desarrollar un sistema de pre-inscripción en línea (página web), que permita a los estudiantes de la Universidad IESAC pre-inscribirse a cada cuatrimestre y facilite en gran medida la consulta de los horarios de las materias a cursar durante el periodo inscrito y de igual manera permita al departamento escolar, agilizar el proceso de re-inscripción debido a la reducción de estudiantes en las instalaciones de la universidad.



1.3. Objetivos Específicos

El portal web implementará los siguientes requerimientos:

1. Diseño y modelado de una base de datos relacional.
2. Implementación de páginas web necesarias que permitan la alta, baja y cambios de cuatrimestres, materias, horarios y profesores de la universidad.
3. Difundir el nuevo proceso de pre-inscripción en la comunidad estudiantil.
4. Notificaciones al estudiantes de su estado administrativo antes de poder pre-inscribirse en línea.
5. Generación de reportes con posibilidad de exportarlo a formato PDF, que contenga la información estadística de los estudiantes pre-inscritos en línea en tiempo real.
6. Disposición de un formulario de Contacto para los estudiantes.

1.4. Justificación

Cuando hablamos del “uso de internet”, abrimos oportunidades en su uso que parecen infinitas y su evolución es tan rápida que mientras lee esta tesina, cientos de personas están innovando para mejorar las funcionalidades y miles se están estrenando como internautas.

Al crecer el número de internautas, crece el número de la comunidad estudiantil en usar la tecnología para acceder al internet, tomando en cuenta este antecedente y la competencia de servicios proporcionados por las universidades privadas, se tiene la necesidad de implementar un servicio tecnológico para ser ofrecido a toda la comunidad estudiantil de la Universidad como un beneficio a los alumnos que cumplan a tiempo con sus cuotas académicas.

El poder usar los servicios tecnológicos de la universidad brinda a los estudiantes un plus extra para poder ser consultado desde cualquier dispositivo fijo o móvil con acceso a internet lo cual implica implícitamente un ahorro en tiempo y dinero para su traslado a las instalaciones de la universidad donde realizar la misma actividad que en el portal web.



El 40% de la matrícula de la Universidad corresponde a estudiantes foráneos, por lo tanto, es conveniente implementar el sistema debido a los beneficios que traerían a este grupo de personas, de igual manera se contribuiría en parte al medio ambiente ya que se podrá evitar el traslado a las instalaciones del 80% de la matrícula a las instalaciones de la universidad.

Otro beneficio importante se vería reflejado en el departamento escolar de la universidad del IESAC, ya que sólo estarán atendiendo en las instalaciones a los estudiantes que tengan una restricción académica que impida su pre-inscripción en línea y una cantidad minoritaria de estudiantes que no hayan accedido al sistema para realizarlo.

1.5. Hipótesis

A medida que avanza la tecnología y el uso de esta en los jóvenes, ha incrementado el acceso a internet, influyendo en la forma y facilidad para la consulta de información vía web, facilitando la comunicación entre usuarios y corporaciones además del uso de servicios personales e institucionales.

2. Marco Teórico

Para llevar a cabo el diseño y el desarrollo del Sistema de Pre-inscripción en línea de la Universidad del IESAC, se llevaron a cabo una serie de procesos para poder obtener la información necesaria para tener una base sólida y con ello un sistema seguro con un buen desempeño. Entre los procesos realizados son:

- Analizar una Metodología Estructurada para desarrollar el Ciclo de Vida completo de los Sistemas Informáticos.
- Informarse de las plataformas y arquitecturas tecnológicas disponibles y según las tendencias y estándares del mercado basados en modelos Cliente / Servidor.
- Conocimiento de Sistemas de Gestión de Bases de Datos y Lenguaje de Consulta Estructurado (MySQL) para el manejo, almacenamiento y procesamiento de los datos.



- Entrevistas con el coordinador de la universidad de la forma de trabajo y los requerimientos del sistema.
- Selección del lenguaje de programación, que sea el más adecuado para la programación que contemple el manejo de Bases de Datos de manera confiable y brinde seguridad en las operaciones del sistema.

2.1. Conceptos Básicos

2.1.1. *Base de Datos*

Es una colección de archivos interrelacionados, creados con un Sistema Manejador de Bases de Datos (DBMS). Esta engloba la información de una organización, de tal manera que los datos estén disponibles para los usuarios. El acceso a esta puede estar clasificada por privilegios de los usuarios, y su principal objetivo es evitar la redundancia de datos o al menos minimizarla.

Los componentes principales en un sistema de base de datos son el hardware, el software DBMS y los datos, así como también al administrador del sistema.

El Administrador de la base de datos DBA, es la persona o equipo de personas responsables del control, manejo y seguridad del sistema de base de datos, con experiencia en el manejo, diseño, comunicación, hardware y programación.

2.1.2. *Modelo Entidad-Relación*

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos, fue introducido por Peter Chen en 1976.

El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido.

En la figura 1 se ilustra la simbología empleada en el modelo Entidad – Relación:

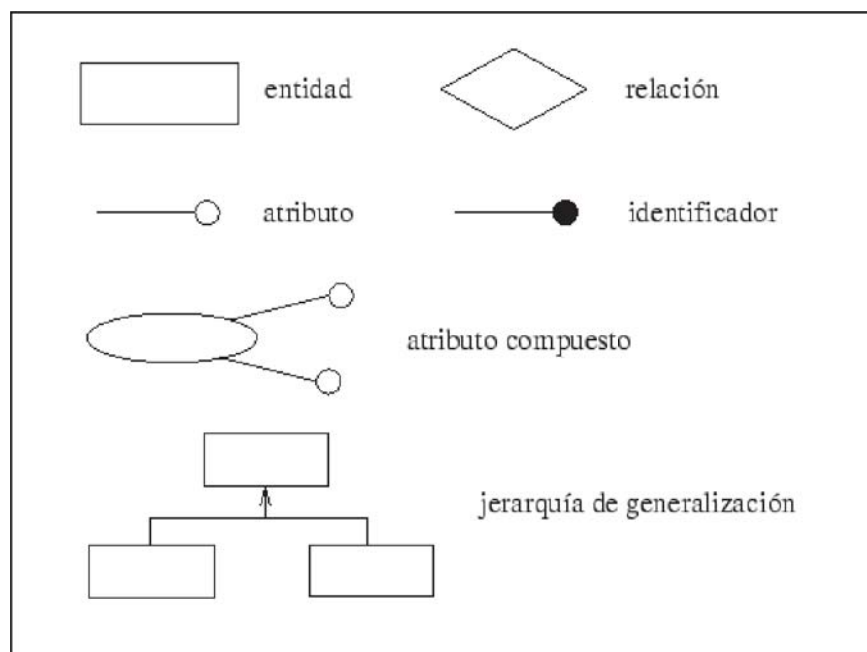


Figura 1: Simbología del Modelo Entidad Relación

2.1.3. Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.



Hay dos tipos de entidades: fuertes y débiles.

- **Entidad débil:** es una entidad cuya existencia depende de la existencia de otra entidad.
- **Entidad fuerte:** es aquella que no necesita otra entidad débil para existir.

2.1.4. *Relación (interrelación)*

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan *entidades participantes*. El número de participantes en una relación es lo que se denomina *grado* de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación *binaria*; si son tres las entidades participantes, la relación es *ternaria*; etc.

Una *relación recursiva* es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es *opcional (parcial)*. Las reglas que definen la cardinalidad de las relaciones son las *reglas de negocio*.



2.1.5. *Atributo*

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante círculos unidos a líneas que le dan sentido a las relaciones.

Cada atributo tiene un conjunto de valores asociados denominado *dominio*. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un *atributo simple* es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un *atributo compuesto* es un atributo con varios componentes, cada uno con un significado por sí mismo.

Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso. Un atributo compuesto se representa gráficamente mediante un óvalo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un *atributo monovalente* es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un *atributo polivalente* es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina *multivaluados*, y pueden tener un número máximo y un número mínimo de valores.

La *cardinalidad* de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es (1, 1).

Por último, los atributos pueden ser derivados. Un *atributo derivado* es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.



2.1.6. Identificador

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos.

2.1.7. Metodología de diseño conceptual

El primer paso en el diseño de una base de datos es la producción del esquema conceptual, normalmente, se construyen varios esquemas conceptuales, cada uno para representar las distintas visiones que los usuarios tienen de la información. Cada una de estas visiones suelen corresponder a las diferentes áreas funcionales de la empresa como, por ejemplo, producción, ventas, recursos humanos, etc.

Estas visiones de la información, denominadas *vistas*, se pueden identificar de varias formas. Una opción consiste en examinar los diagramas de flujo de datos, que se pueden haber producido previamente, para identificar cada una de las áreas funcionales. La otra opción consiste en entrevistar a los usuarios, examinar los procedimientos, los informes y los formularios, y también observar el funcionamiento de la empresa.

A los esquemas conceptuales correspondientes a cada vista de usuario se les denomina *esquemas conceptuales locales*. Cada uno de estos esquemas se compone de entidades, relaciones, atributos, dominios de atributos e identificadores. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo.



Las tareas a realizar en el diseño conceptual son las siguientes:

1. Identificar las entidades.
2. Identificar las relaciones.
3. Identificar los atributos y asociarlos a entidades y relaciones.
4. Determinar los dominios de los atributos.
5. Determinar los identificadores.
6. Determinar las jerarquías de generalización (si las hay).
7. Dibujar el diagrama entidad-relación.
8. Revisar el esquema conceptual local con el usuario.

2.1.8. Normalización

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad.

Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

Uno de los conceptos fundamentales en la normalización es el de *dependencia funcional*. Una *dependencia funcional* es una relación entre atributos de una misma relación (tabla). Si x e y son atributos de la relación R , se dice que y es funcionalmente dependiente de x (se denota por $x \rightarrow y$) si cada valor de x tiene asociado un solo valor de y (x e y pueden constar de uno o varios atributos). A x se le denomina *determinante*, ya que x determina el valor de y . Se dice que el atributo y es *completamente dependiente* de x si depende funcionalmente de x y no depende de ningún subconjunto de x .



La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad.

En el proceso de normalización se debe ir comprobando que cada relación (tabla) cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

Primera forma normal (1FN)

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos. En la primera, se repiten los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales.



La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

Segunda forma normal (2FN)

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo no primo (que no está en la clave primaria) es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).

Tercera forma normal (3FN)

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo no primo no depende transitivamente de la clave primaria. La dependencia $x \rightarrow z$ es transitiva si existen las dependencias $x \rightarrow y$, $y \rightarrow z$, siendo x , y , atributos o conjuntos de atributos de una misma relación.

Aunque las relaciones en 2FN tienen menos redundancias que las relaciones en 1FN, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen).



Forma normal de Boyce-Codd (BCFN)

Una relación está en la forma normal de Boyce-Codd si, y sólo si, todo determinante es una clave candidata.

La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si éstas existen. La BCFN es más fuerte que la 3FN, por lo tanto, toda relación en BCFN está en 3FN.

La violación de la BCFN es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación viola la BCFN si tiene dos o más claves candidatas compuestas que tienen al menos un atributo en común.

2.2. Sistema de Gestión de Bases de Datos (DBMS Database Management System)

Un sistema de gestión de bases de datos (DBMS Database Management System) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos, es básicamente un sistema computarizado para llevar registros, cuya finalidad general es almacenar información y permitir a los usuarios realizar una variedad de operaciones sobre dichos registros, por ejemplo: insertar, borrar, modificar, agregar, consultar. La colección de datos se denomina base de datos (BD).

El objetivo primordial de un DBMS es poder extraer o almacenar información en la BD de forma eficiente.

Los sistemas de bases de datos están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para el almacenamiento como de mecanismos para la gestión de la información.

Además los DBMS deben mantener la seguridad de la información almacenada pese a la caída del sistema o accesos no autorizados.



Los objetivos principales de un Sistema de Gestión de Bases de Datos es disminuir los siguientes aspectos:

2.2.1. Redundancia e inconsistencia de datos.

Los datos al ser manipulados por diferentes tipos de programas existe la posibilidad de que si no se controla el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información este repetida más de una vez en un dispositivo de almacenamiento, provocando inconsistencia en los datos.

2.2.2. Dificultad para tener acceso a los datos.

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos. El sistema debe de prever estas situaciones desde que se realiza el diseño para evitar una deficiencia.

2.2.3. Aislamiento de los datos.

Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

2.2.4. Anomalías del acceso concurrente.

Los sistemas pueden realizar operaciones simultáneas sobres los datos, con la finalidad de mejorar el funcionamiento y la respuesta sea más rápida y eficiente. En un entorno operativo las actualizaciones concurrentes pueden dar por resultado datos inconsistentes.



2.2.5. *Problemas de seguridad.*

La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la consulta, autenticación y protección de los datos.

2.2.6. *Problemas de integridad.*

Los valores de datos almacenados en la base de datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación.

http://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_2.htm

2.3. Acceso a Base de Datos

2.3.1. *PDO (PHP Data Objects)*

La extensión de objetos de datos de PHP (PDO, *Objetos de Datos de PHP*), proporciona una interfaz unificada para trabajar con bases de datos y ayuda a que los desarrolladores manipulen diferentes bases de datos de manera consistente.

PDO dispone de diferentes controladores de bases de datos, ofreciendo unas características muy importantes, como la portabilidad y consistencia para diferentes motores de bases de datos, así como un nivel alto de seguridad y desempeño.

PDO proporciona una capa de abstracción de *acceso a datos*, lo que significa que, independientemente de la base de datos que se esté utilizando, se usan las mismas funciones para realizar consultas y obtener datos. PDO *no* proporciona una abstracción de *bases de datos*; no reescribe SQL ni emula características ausentes. Se debería usar una capa de abstracción totalmente desarrollada si fuera necesaria tal capacidad.

<http://www.php.net/manual/es/intro.pdo.php>



2.4. Conceptos Básicos Web

2.4.1. *Cliente - Servidor*

La programación cliente-servidor es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere el servicio de una máquina servidor, y éste realiza la función para la que está programado, como por ejemplo: servidores Web, servidor de archivos, servidores de correo etc.

La separación entre cliente y servidor es de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

El termino cliente / servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o más programas: una aplicación y un servicio soportante.

Los elementos principales de la arquitectura cliente / servidor son justamente el elemento llamado cliente y el otro elemento llamado servidor. Ambas partes de la aplicación se comunican entre sí utilizando el protocolo de red TCP/IP.

El término Cliente/Servidor fue usado por primera vez en 1980 para referirse a PC's en red, y empezó a ser aceptado a finales de los 80's.

Desde el punto de vista funcional, se puede decir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener información en forma transparente aún en entornos multiplataforma.

Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco etc. La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.

En la figura 2 se muestra gráficamente el funcionamiento del modelo Cliente / Servidor:

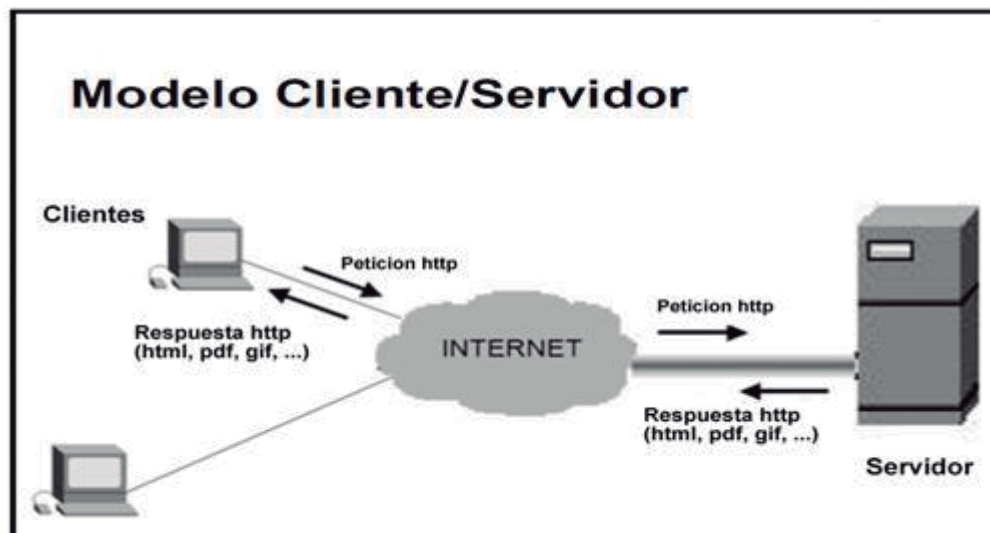


Figura 2: Modelo Cliente - Servidor

2.4.2. Servidores Web

Un servidor web es un programa que se ejecuta continuamente en un computador, y la principal función es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios. Básicamente, un servidor Web es una gran computadora que guarda y transmite datos vía Internet. Cuando un usuario entra en una página de Internet su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso decimos que los servidores Web están para almacenar y transmitir datos de un sitio según lo que pida el navegador de un visitante.

Cada servidor Web y cada computadora conectada a Internet tienen asignado una dirección de IP irreplicable que lo identifica en la red incluyendo la PC.



Cuando se consulta una página Web, se envía una petición desde la dirección IP del cliente hacia la dirección IP del servidor. El servidor Web responde mandando datos a la dirección IP que los pide, esta forma de peticiones y respuestas se repite cada vez que se navega en un sitio en Internet.

Más del 90% de los sitios en Internet utilizan un servidor alquilado a través de una empresa de Web hosting.

2.4.3. Apache

Apache es uno de los Servidores de páginas más utilizados, al ofrecer instalaciones sencillas para sitios pequeños y si se requiere es posible expandirlo hasta el nivel de los mejores productos comerciales.

Entre las principales características de Apache, es que es personalizable y su arquitectura es modular permitiendo construir un servidor hecho a la medida.

En cuanto a la administración de los archivos de configuración de Apache están en ASCII, por lo que tiene un formato simple, y pueden ser editados con un editor de texto, siendo estos transferibles por lo que la portabilidad de éste es sencilla.

Cuando el Servidor de Páginas Web recibe la petición de un cliente, éste reconoce cuando debe enviar un documento estático o ejecutar algún tipo de aplicación, por ejemplo: si se invoca un programa en Perl y este a su vez solicita información a una base de datos, por lo tanto para llevar a cabo esta operación se inician dos procesos nuevos.

Apache tiene tanto tiempo de desarrollo que han sido desarrolladas diferentes soluciones para evitar ineficiencias, siendo su código en Lenguaje C, ha permitido realizar optimizaciones a éste, haciendo rápido y que consuma menos recursos del sistema en comparación con otros servidores. Apache corre en una amplia variedad de Sistemas Operativos (OS), incluyendo varias versiones de Unix, Windows y MacOS.



2.4.4. Seguridad

La evaluación de este punto es uno de los más importantes en la interconexión de la página Web con las bases de datos. Para conservar la integridad de los datos y de las estructuras se deben crear usuarios que puedan realizar determinadas tareas, y que otras solo puedan realizarse por un número limitado y controlado de usuarios.

Los conceptos de usuarios y privilegios están íntimamente relacionados. No se pueden crear usuarios sin asignarle al mismo tiempo privilegios. De hecho, la necesidad de crear usuarios está ligada a la necesidad de limitar las acciones que tales usuarios pueden llevar a cabo.

El uso de reCaptchas en los sistemas, brindan una mayor seguridad al emplear la prueba desafío-respuesta, utilizada en computación para determinar cuando el usuario es o no humano, siendo una tarea simple para un ser humano pero para una máquina resulta ser una tarea bastante compleja.

El uso de ReCaptchas consiste en sustituir al sistema Captcha, colocando dos palabras a reconocer (en lugar de una que emplea típicamente la prueba Captcha). Una de las palabras es conocida y la otra es desconocida para el sistema. La palabra desconocida es una que no pudo ser obtenida de una imagen mediante un sistema OCR automatizado. El sistema pide al usuario (quien desconoce qué palabra es conocida y cuál no lo es) que introduzca ambas palabras como texto. Si la palabra conocida por el sistema es introducida correctamente por un humano, el sistema de reCaptcha asume que hay probabilidades altas de que el usuario también haya introducido la palabra desconocida correctamente. Si la palabra desconocida recibe en múltiples ocasiones la misma transcripción humana (traducción de imagen a texto) se considera que esa transcripción es correcta.

ReCaptcha es una herramienta de Google que usa un avanzado motor de análisis de riesgos y de adaptación de Captchas para restringir el acceso al software automatizado en el sitio, mientras deja que los usuarios válidos ingresen con facilidad, protegiendo a los sitios web del spam y el fraude.

<http://repositorio.utc.edu.ec/bitstream/27000/11111/1/T-UTC-0772.pdf>



2.4.5. Hipertext Preprocesor (PHP)

Hipertext Preprocesor (PHP) es un lenguaje de programación que se ejecuta del lado del servidor con una gran librería de funciones, que se incrustan dentro del código HTML de una página web.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al navegador Web. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. (Ver figura 3). Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad.

Su plataforma es independiente, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto. A lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red.

Vikram Vaswani (2009), Fundamentos de Php

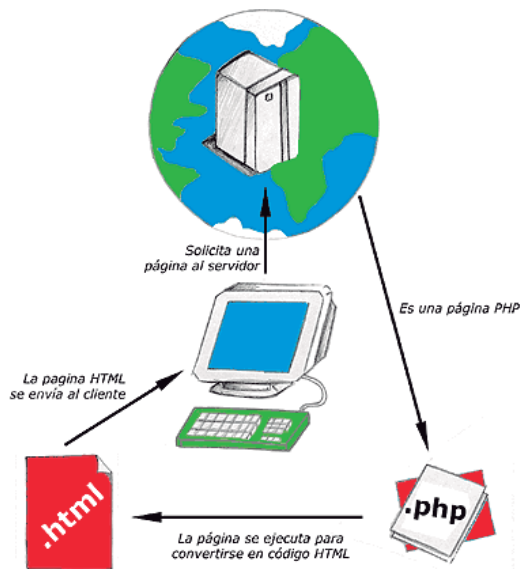


Figura 3: Esquema del Funcionamiento de las páginas PHP

2.4.6. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario y operaciones que desencadenan llamadas al servidor, que una vez procesadas, el servidor devuelve una información que puede ser una nueva página siendo este proceso transparente para el usuario final.

Técnicamente, JavaScript es un lenguaje de programación interpretado, es decir, que es ejecutado del lado del cliente, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.



2.4.7. JQuery

JQuery es una biblioteca gratuita de JavaScript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acordes a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos.

El uso de JQuery en los sitios web ayuda a crear efectos dinámicos, manipulación asíncrona mediante Ajax, interfaces de usuario avanzados, etc., la principal ventaja del uso esta librería es que ya se encuentran implementadas y probadas, lo cual permitirá una programación más rápida y libre de errores.

3. Metodología de Investigación

El modelo planteado para organizar el proceso de desarrollo del sistema, son una serie de actividades que determinan en orden de la etapas involucradas y los criterios de transacción asociadas entre las etapas.

El modelo de ingeniería de software de esta tesina sigue el Modelo Lineal Secuencial, que será descrito a continuación:

El **Modelo Lineal Secuencial**, también conocido como modelo en cascada, se basa en un enfoque sistemático y secuencial del desarrollo del software que comienza en un nivel de Ingeniería y modelado del sistema, análisis, diseño, codificación, pruebas y mantenimiento.

La siguiente figura ilustra el modelo lineal secuencial para la ingeniería de software.

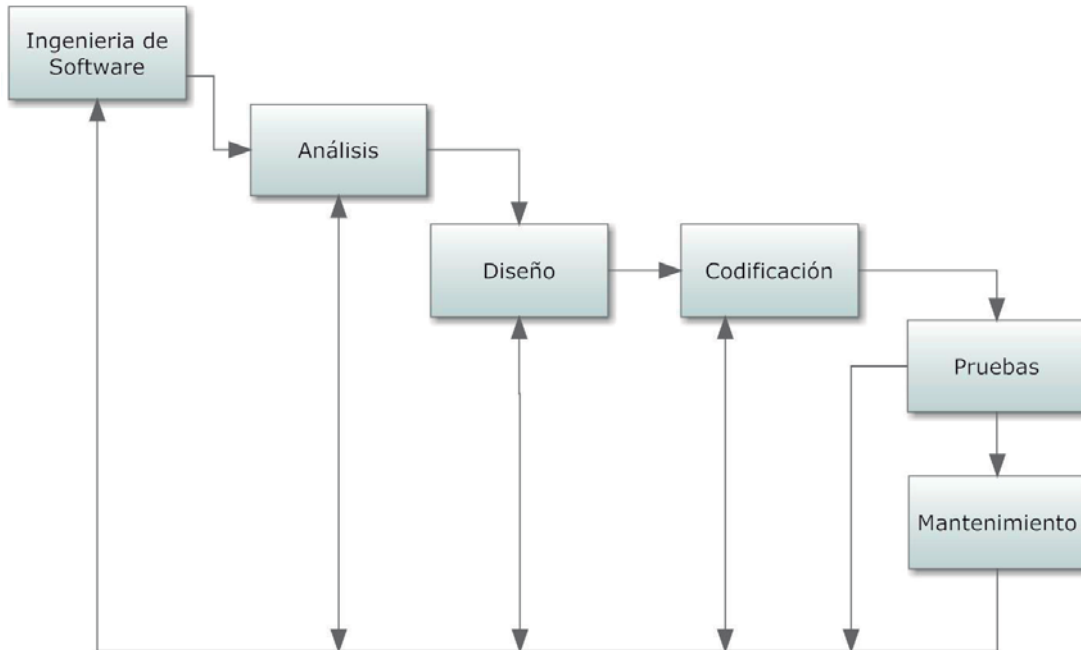


Figura 4 : Modelo Lineal Secuencial (Modelo en Cascada)

De acuerdo a Pressman, el modelo lineal secuencia contempla seis actividades que deben llevarse a cabo. A continuación se describen cada una de estas actividades:

Ingeniería y análisis del sistema.

En general todo proceso de ingeniería debe comenzar contestando las siguientes preguntas: ¿Cuál es el problema a resolver? ¿Qué características debe tener el sistema para resolver el problema?, ¿Cómo se realizará el sistema?, ¿Cómo va a probarse el sistema? Y ¿Cómo se atenderá las peticiones de los usuario cuando soliciten correcciones y adaptaciones al sistema?,

Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo todos los requerimientos o elementos del sistema y luego asignando algún subconjunto de estos requerimientos al software; esta versión del sistema es esencial cuando el software debe interrelacionarse con otros elementos tales como hardware, personas y bases de datos.



La ingeniería y análisis del sistema abarcan los requerimientos globales a un nivel de sistema con una pequeña cantidad de análisis y diseño a nivel superior. Además de un análisis costo beneficio del sistema es decir si toda la inversión que se hará para el sistema conviene a los beneficios que traerá el mismo.

Análisis de los requerimientos del software.

El proceso de recoger los requerimientos se centra y se intensifica especialmente en esta etapa, para comprender completamente la naturaleza de los programas que se deben desarrollar para la aplicación. El ingeniero de software debe comprender el dominio de la información del software, así como la función, rendimiento e interfaces requeridas. En esta etapa los requerimientos del sistema se documentan y se analizan con el cliente.

Diseño

El diseño del software es realmente un proceso de muchos pasos refiriéndose al establecimiento de la estructura de datos, la arquitectura general del software e interfaces del sistema.

El diseño traduce los requerimientos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes que comience la codificación. Como los requerimientos y el diseño que se documentan forman parte de la configuración del software.

Codificación.

El paso de la codificación consiste en traducir el diseño de una forma legible para la máquina. En el caso de la codificación de este proyecto, se refiere tanto al desarrollo del sistema como de la configuración del sitio donde se alojará la página web en la cual se añadirá comportamiento en estos ambientes.



Prueba.

Una vez que se ha generado el código, comienza la prueba del programa, la prueba se enfoca sobre la lógica interna del software asegurando que todas las sentencias se han comprobado y las funciones externas se realicen para la detección de errores.

El software requiere una etapa de pruebas con el(los) usuario(s) final(es) del sistema, con la finalidad de proporcionar retroalimentación a los desarrolladores, sin perder el contacto con el solicitante del sistema, manteniendo los objetivos del proyecto siempre vigentes y una idea clara de los aspectos que se deben probar en el periodo de prueba.

Mantenimiento.

El software sufrirá indudablemente cambios después que se le entregue al cliente; los cambios ocurrirán debido a que el cliente ha encontrado mejoras y por lo tanto el aumento de funciones al sistema.

El mantenimiento del software se aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en lugar de uno nuevo, y si existe la documentación apropiada, esta puede utilizarse posteriormente como referencia.



3.1. Elementos de UML

Diagrama de casos de uso

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo.

Caso de uso

Un caso de uso describe, desde el punto de vista de los actores, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener este (recuerde, únicamente el qué, nunca el cómo).

Cuando se trabaja con casos de uso, es importante tener presentes algunas sencillas reglas:

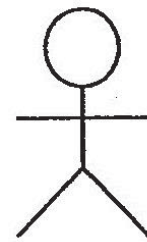
- Cada caso de uso está relacionado como mínimo con un actor
- Cada caso de uso es un iniciador (es decir, un actor)
- Cada caso de uso lleva a un resultado relevante (un resultado con «valor intrínseco»)

Los casos de uso pueden tener relaciones con otros casos de uso. Los tres tipos de relaciones más comunes entre casos de uso son:

- <<include>> que especifica una situación en la que un caso de uso tiene lugar dentro de otro caso de uso
- <<extends>> que especifica que en ciertas situaciones, o en algún punto (llamado punto de extensión) un caso de uso será extendido por otro.
- Generalización que especifica que un caso de uso hereda las características del «super» caso de uso, y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.



Actor

Los actores no representan a personas físicas o a sistemas, sino su rol. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. Por ejemplo, una persona que proporciona servicios de atención telefónica a clientes y realiza pedidos para los clientes estaría representada por un actor «equipo de soporte» y por otro actor «representante de ventas».

Descripción de casos de uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

Diagrama de clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Clase

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objeto tiene el suyo propio). En ocasiones se utiliza el término «tipo» en lugar de clase, pero recuerde que no son lo mismo, y que el término tipo tiene un significado más general.

Las clases están representadas por rectángulos, con el nombre de la clase, y también pueden mostrar atributos y operaciones de la clase en otros dos «compartimentos» dentro del rectángulo, como se muestra en la figura 5.

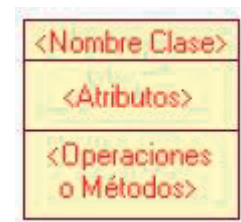


Figura 5: Clase

Atributos

En UML, los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades. Los atributos también pueden ser mostrados visualmente:

- + Indica atributos *públicos*
- # Indica atributos *protegidos*
- - Indica atributos *privados*

Operaciones

Las operaciones (métodos) también se muestran al menos con su nombre, y pueden mostrar sus parámetros y valores de retorno. Las operaciones, al igual que los atributos, se pueden mostrar visualmente:

- + Indica operaciones *públicas*
- # Indica operaciones *protegidas*
- - Indica operaciones *privadas*

Asociaciones de clases

Las clases se puede relaciones (estar asociadas) con otras de diferentes maneras:

Generalización

La herencia es uno de los conceptos fundamentales de la programación orientada a objetos, en la que una clase «recoge» todos los atributos y operaciones de la clase de la que es heredera, y puede alterar/modificar algunos de ellos, así como añadir más atributos y operaciones propias.

En UML, una asociación de *generalización* entre dos clases, coloca a estas en una jerarquía que representa el concepto de herencia de una clase derivada de la clase base. En UML, las generalizaciones se representan por medio de una línea que conecta las dos clases, con una flecha en el lado de la clase base, como se ilustra en la figura 6.

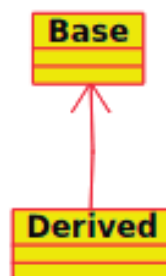


Figura 6 Representación visual de una generalización en UML

Asociaciones

Una asociación representa una relación entre clases, y aporta la semántica común y la estructura de muchos tipos de «conexiones» entre objetos.

Las asociaciones son los mecanismos que permite a los objetos comunicarse entre sí. Describe la conexión entre diferentes clases (la conexión entre los objetos reales se denomina conexión de objetos o *enlace*).

Las asociaciones pueden tener un papel que especifica el propósito de la asociación y pueden ser unidireccionales o bidireccionales (indicando si los dos objetos participantes en la relación pueden intercambiar mensajes entre sí, o es únicamente uno de ellos el que recibe información del otro). Cada extremo de la asociación también tiene un valor de multiplicidad, que indica cuántos objetos de ese lado de la asociación están relacionados con un objeto del extremo contrario.

En UML, las asociaciones se representan por medio de líneas que conectan las clases participantes en la relación, y también pueden mostrar el papel y la multiplicidad de cada uno de los participantes. La multiplicidad se muestra como un rango [mín...máx] de valores no negativos, con un asterisco (*) representando el infinito en el lado máximo, como se muestra en la figura



Figura 7 Representación visual de una asociación en UML

Acumulación

Las acumulaciones son tipos especiales de asociaciones en las que las dos clases participantes no tienen un estado igual, pero constituyen una relación «completa». Una acumulación describe cómo se compone la clase que asume el rol completo de otras clases que se encargan de las partes. En las acumulaciones, la clase que actúa como completa, tiene una multiplicidad de uno.

En UML, las acumulaciones están representadas por una asociación que muestra un rombo en uno de los lados de la clase completa, como se ilustra en la siguiente figura 8.



Figura 8 Representación visual de una relación de acumulación en UML

Composición

Las composiciones son asociaciones que representan acumulaciones *muy fuertes*. Esto significa que las composiciones también forman relaciones completas, pero dichas relaciones son tan fuertes que las partes no pueden existir por sí mismas. Únicamente existen como parte del conjunto, y si este es destruido las partes también lo son.

En UML, las composiciones están representadas por un rombo sólido al lado del conjunto, como se muestra en la figura 9.



Figura 9 Representación visual de una relación de acumulación en UML

Otros componentes de los diagramas de clases

Los diagramas de clases pueden contener más componentes aparte de clases.

Interfaces

Las interfaces son clases abstractas, lo que significa que no es posible crear instancias directamente a partir de ellas. Pueden contener operaciones, pero no atributos. Las clases pueden heredar de las interfaces (a través de una asociación de realización) y de estos diagramas sí es posible crear instancias.



Tipo de datos

Los tipos de datos son primitivas construidas normalmente en algunos lenguajes de programación. Algunos ejemplos comunes son los enteros y los booleanos. No pueden tener relación con clases, pero las clases sí pueden relacionarse con ellos.

Enumeraciones

Las enumeraciones son simples listas de valores. Un ejemplo típico de esto sería una enumeración de los días de la semana. Las opciones de una enumeración se llaman «literales de enumeración». Al igual que los tipos de datos, no pueden relacionarse con las clases, pero las clases sí pueden hacerlo con ellos.

Paquetes

Los paquetes, en lenguajes de programación, representan un espacio de nombres en un diagrama se emplean para representar partes del sistema que contienen más de una clase, incluso cientos de ellas.

Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

<http://docs.kde.org/stable/es/kdesdk/umbrello/uml-elements.html>



4. Propuesta del Sistema

En este capítulo se debe abordar ampliamente el diseño e implementación del tema de tesis. Basándose en los requerimientos, se describe de forma clara los métodos, técnicas, procedimientos y demás herramientas utilizadas para el desarrollo del sistema, así como para la implementación de la misma.

4.1. Análisis y Diseño

4.1.1. Requerimientos Funcionales

Código	Descripción	Prioridad
RF : 1	El sistema debe permitir solo el acceso de los alumnos del IESAC y al Coordinador de la Universidad, valiéndose de un nombre de usuario (matricula) y contraseña.	Alta
RF : 2	El sistema permitirá realizar el registro de Cuatrimestres por periodo (primavera, verano y otoño) de cada carrera de la Universidad.	Alta
RF : 3	El sistema facilitará la captura de materias por cuatrimestre, donde cada materia deberá tener asignado un profesor, una hora inicial, una hora final, un salón y los días de la semana que se impartirá.	Alta
RF: 4	El coordinador de la universidad (administrador) podrá consultar, modificar y registrar nuevas materias, profesores, salones y periodos de pre-inscripción.	Alta
RF : 5	El sistema brindará información al estudiante de su estado académico, además de permitirle pre-inscribirse en línea en caso de no contar con una restricción académica.	Media
RF : 6	El sistema facilitará al estudiante la consulta en línea y descargar del horario de materias del cuatrimestre en formato <u>PDF</u>	Alta
RF 7	El sistema permitirá generar reportes del número de estudiantes pre-inscritos en tiempo real.	Media

4.1.2. Requerimientos No Funcionales

Código	Descripción	Prioridad
RNF : 1	El sistema debe ser accesible desde internet y poder visualizarse correctamente en cualquier navegador web	Alta
RNF : 2	El sistema debe cumplir las disposiciones recogidas en la Ley de datos personales y medidas de seguridad.	Alta
RNF : 3	El sistema no debe tardar más de 5 segundo en generar el horario de materias del cuatrimestre.	Alta

4.1.3. Descripción General del Sistema

4.1.3.1. Diagrama de Flujo de Datos a Nivel Contexto



Figura 10: Diagrama de Flujo de datos a Nivel Contexto

4.1.3.2. Casos de Uso

ESCENARIO DE CASO DE USO

COORDINADOR DE LA UNIVERSIDAD

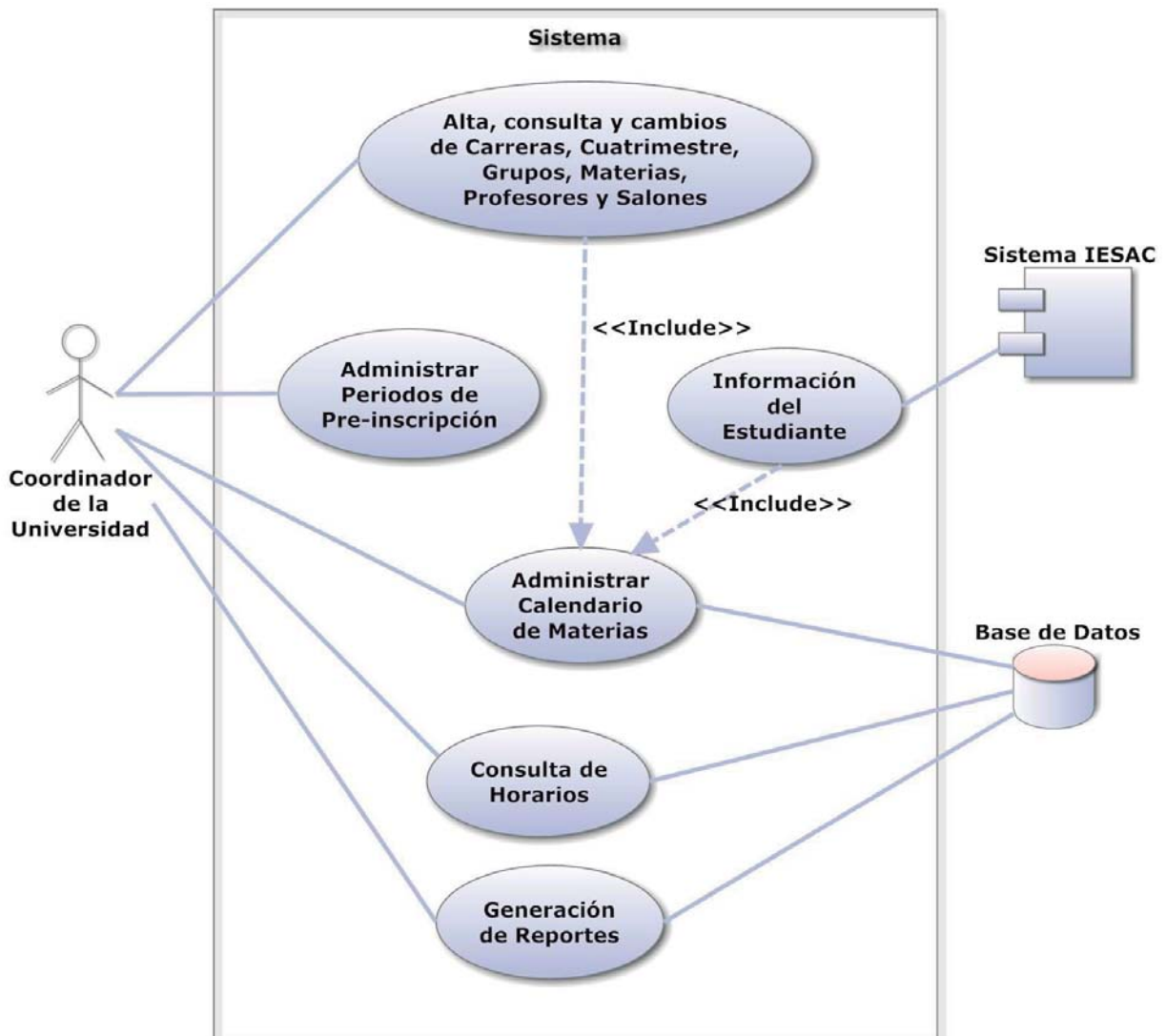


Figura 11: Caso de uso Coordinador de la Universidad

ESCENARIO DE CASO DE USO

ESTUDIANTE

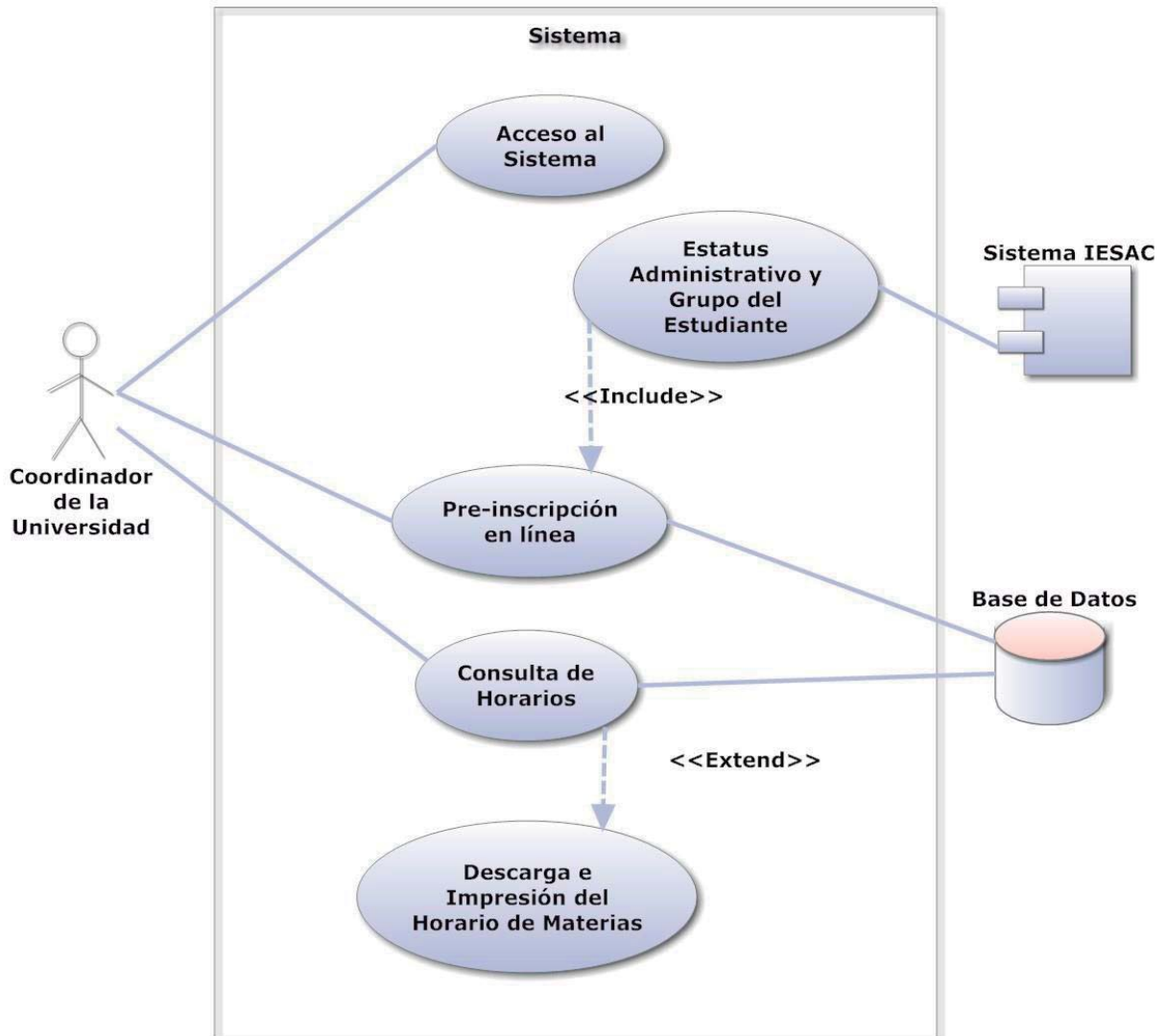


Figura 12: Caso de uso Estudiante



4.1.3.3. Escenarios

Escenario 1 Caso de uso 1 Alta, Consulta y Cambios de Carreras, Cuatrimestres, Grupos, Materias, Profesores y Salones

Nombre del Caso de Uso :	Alta, Consulta y Cambios de Carreras, Cuatrimestres, Grupos, Materias, Profesores y Salones	
Actor:	Coordinador de la Universidad	
Descripción:	El Coordinador de la Universidad podrá registrar y modificar las carreras y materias impartidas en la Universidad, así como también a los profesores que las imparten, el cuatrimestre que estará vigente para la pre-inscripción y los salones asignados para las clases.	
Evento Desencadenador:	El Coordinador de la Universidad ingresa al sitio web identificándose con un nombre de usuario y contraseña y da clic en la opción “Registrar” o “Actualizar” del menú de navegación, según sea el caso.	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El Coordinador de la Universidad inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña.	
2. El sistema muestra el menú de navegación y selecciona la opción “Registro” o “Actualizar” según sea el caso.	Menú de navegación	
3. El sistema despliega un menú de opciones (carrera, cuatrimestre, grupo, materia, profesor o salón).	Página web de registro o actualización	
4. Se seleccionar alguna opción del menú y se llena o modifica con la información solicitada por el sistema y se da clic en el botón “Aceptar”.	Información para el llenado de los Formularios.	
5. Se valida la información introducida en los campos del formulario y se procede con el registro en la base de datos.	Conexión con la base de datos.	
Precondiciones:	<p>Debe disponer de una conexión a internet.</p> <p>El usuario deberá iniciar sesión en el sistema y contar con los privilegios para realizar esta actividad.</p> <p>Contar con la información necesaria para el registro.</p>	
Garantía de Éxito:	El Coordinador de la Universidad (usuario) registró o modificó con éxito una Carrera, Cuatrimestre, Grupo, Materia, Profesores o Salón.	



Escenario 1 Caso de uso 2

Administrar Periodos de Pre-Inscripción

Nombre del Caso de uso:	Administrar Periodos de Pre-Inscripción
Actor:	Coordinador de la Universidad
Descripción:	El Coordinador de la Universidad podrá administrar los periodos de pre-inscripción registrando el año, la fecha y hora inicial y final que estará abierto el sistema a los estudiantes para poder pre-inscribirse en línea.
Evento Desencadenador:	El Coordinador de la Universidad después de haber ingresado al sistema, da clic en la opción “Crear Pre-inscripción” del menú de navegación.
Pasos Realizados (ruta principal):	Información para los pasos:
1. El Coordinador de la Universidad inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña.
2. El sistema muestra el menú de navegación y selecciona la opción “Crear Pre-inscripción”.	Menú de navegación
3. Para registrar un nuevo periodo de pre-inscripción se da clic en el botón “Nuevo Periodo” y se llenan todos los campos el formulario de registro, al término dar clic en el botón “Aceptar”.	Información del periodo de pre-inscripción.
4. Para modificar un periodo de pre-inscripción se selecciona un periodo de la lista desplegable y se da clic en el botón “Buscar”.	Formulario web de registro
5. La información del periodo se carga en formulario para su modificación, para aceptar los cambios realizados dar clic en el botón “Actualizar Periodo”	Información del periodo de pre-inscripción.
6. Antes de registrar o actualizar la información de los periodos de pre-inscripción en la base de datos, se lleva a cabo una validación de la información de los campos.	Página web de registro de cursos
Precondiciones:	El usuario deberá iniciar sesión en el sistema y contar con los privilegios para realizar esta actividad y Contar con la información necesaria para el registro de los periodos de pre-inscripción.
Garantía de Éxito:	El coordinador de la universidad registró y/o actualizó la información de los periodos de pre-inscripción.



Escenario 1 Caso de uso 3 Administrar Calendario de Materias

Nombre del Caso:	Administrar Calendario de Materias	
Actor:	Coordinador de la Universidad	
Descripción:	Cada estudiante de la universidad pertenece a un grupo según la carrera y el periodo que cursa, el Coordinador de la Universidad podrá crear el calendario de materias que se impartirán en el cuatrimestre.	
Evento Desencadenador:	El Coordinador ingresa al sitio web identificándose con su nombre de usuario y contraseña y da clic en la opción “Registrar” del menú de navegación y posteriormente en la opción “Curso”.	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El Coordinador de la Universidad inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña.	
2. El sistema muestra el menú de navegación y se selecciona la opción “Registrar”, se desplegará un menú de opciones y dar clic en la opción “Curso”.	Menú de navegación	
3. Se desplegará un formulario de registro donde se seleccionará necesariamente la carrera, el grupo, cuatrimestre y año que corresponde al horario de materias	Formulario de registro	
4. El sistema permite de forma dinámica administrar la carga de materias, con posibilidad de asignar el profesor que la impartirá, horario y día de la semana que se dará la clase.	Formulario de registro	
5. Al finalizar la creación de horario de materias, se da clic en el botón “Registrar”, el sistema valida la congruencia de horarios y días de la semana que serán impartidas las clases para evitar cruce entre materias.	Formulario web de registro	
6. Se registra el horario de materias que se impartirán en el cuatrimestre.	Conexión a Base de Datos	
Precondiciones:	El usuario deberá contar con los privilegios necesarios para realizar esta actividad y deberá existir al menos una carrera, un grupo, un cuatrimestre vigente y una materia para crear el horario.	
Garantía de Éxito:	El coordinador de la universidad crea el horario de materias de todas las carreras del IESAC de acuerdo al cuatrimestre a cursar.	



Escenario 1 Caso de Uso 4 Consulta de Horarios

Nombre del Caso de uso:	Consulta de Horarios	
Actor:	Coordinador de la Universidad	
Descripción:	El coordinador de la Universidad podrá consultar los horarios de materias de todas las carreras de la universidad.	
Evento Desencadenador:	El Coordinador de la Universidad ingresa al sitio web identificándose con un nombre de usuario y contraseña y da clic en la opción "Consultar Horarios" del menú de navegación.	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El Coordinador de la Universidad inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña	
2. El sistema muestra el menú de navegación y selecciona la opción "Consultar Horarios".	Menú de navegación	
3. Se despliega el formulario de consulta, donde se selecciona la carrera y el cuatrimestre, dar clic en el botón "Aceptar".	Formulario web de Consulta	
4. El sistema consulta el horario según las opciones seleccionados por el usuario.	Conexión con la base de datos	
5. Se muestra el horario de materias del cuatrimestre con opción de exportarlo a formato PDF. El horario tiene como columnas los días de la semana y como filas la hora en que se imparte la materia.	Página web y Librería de Php para la generación de archivos en PDF.	
Precondiciones:	El usuario deberá contar con los privilegios necesarios para realizar esta actividad. Debe estar registrado un horario de materias en la base de datos.	
Garantía de Éxito:	El Coordinador de la Universidad consultó todos los horarios de las carreras del cuatrimestre y los exportó en formato PDF.	



Escenario 1 Caso de uso 5 Generar Reportes

Nombre del Caso de uso:	Generar Reportes	
Actor(es):	Coordinador de la Universidad	
Descripción:	El Coordinador de la Universidad podrá generar diferentes reportes informativos acerca del proceso de pre-inscripción en línea.	
Evento Desencadenador:	El administrador ingresa al sitio web identificándose con un nombre de usuario y contraseña y da clic en la opción “Reportes” del menú de navegación.	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El Coordinador de la Universidad inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña.	
2. El sistema muestra el menú de navegación y se selecciona la opción “Reportes”.	Menú de navegación	
3. Se despliega un listado de reportes predefinidos por el Coordinador de la Universidad.	Formulario de consulta	
4. Se selecciona un reporte y se da clic en el botón “Consultar”.	Formulario de consulta	
5. El sistema realiza la consulta y muestra el resultado de manera informativa, éste se podrá exportar en formato PDF o a Microsoft Excel para su impresión o uso administrativo.	Librería de Php para la generación de archivos en Excel y PDF.	
Precondiciones:	El usuario deberá contar con los privilegios necesarios para realizar esta actividad. La librería de Php para la generación de archivos en Excel y PDF deberá estar configurada correctamente.	
Garantía de Éxito:	Consulta de reportes y su exportación de estos a formato PDF y Microsoft Excel.	
Garantía Mínima:	Consulta de reportes en línea.	



Escenario 2 Caso de uso 1

Acceso al Sistema

Nombre del Caso de uso:	Acceso al Sistema	
Actor(es):	Estudiante	
Descripción:	El estudiante podrá ingresar al sistema con su nombre de usuario y contraseña para consultar su estado administrativo, pre-inscribirse en línea y consultar o descargar su horario de materias.	
Evento Desencadenador:	El estudiante ingresa al sitio web, introduce su nombre de usuario (matrícula) y contraseña y da clic en el botón "Aceptar".	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El estudiante inicia sesión en el sitio web ingresando su nombre de usuario (matrícula), contraseña y reCaptcha. Da clic en el botón "Aceptar".	Formulario de acceso Nombre de usuario y contraseña	
2. Se valida el nombre de usuario y contraseña en la base de datos y se registra la sesión.	Conexión con la base de datos	
3. El sistema consulta el tipo de usuario que ingresa para mostrar el menú de navegación correspondiente.	Conexión con la base de datos	
Precondiciones:	Debe disponer con acceso a internet. El estudiante debe estar registrado con su número de matrícula en la base de datos del IESAC. El estatus de la matrícula del estudiante debe estar activa.	
Garantía de Éxito:	El estudiante puede ingresar al sitio web.	



Escenario 2 Caso de uso 2

Pre-inscripción en línea

Nombre del Caso de uso:	Pre-inscripción en línea	
Actor(es):	Estudiante	
Descripción:	El estudiante podrá consulta su estado administrativo para poder pre-inscribirse en línea.	
Evento Desencadenador:	El estudiante ingresa al sitio web, introduce su nombre de usuario (matrícula) y contraseña y selecciona la opción “Consultar Estado Académico” del menú de navegación.	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El estudiante inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña	
2. El sistema muestra el menú de navegación y se selecciona la opción “Consultar Estado Académico”.	Menú de navegación	
3. El sistema consulta el estatus académico del alumno desplegando éste, en la misma página.	Conexión con la base de datos	
4. Si el estatus académico del alumno permite la pre-inscripción en línea, se muestra el botón “Pre-Inscripción en línea”, dar clic sobre éste.	Formulario de consulta.	
5. Se confirma el periodo (cuatrimestre) a pre-inscribir y dar clic en el botón “Aceptar”.	Formulario de pre-inscripción.	
6. Se confirma la pre-inscripción al cuatrimestre para su registro en la base de datos.	Conexión con la base de datos	
7. Se notifica al estudiante que fue pre-inscripción en línea correctamente.		
Precondiciones:	El estudiante debe estar registrado en la base de datos, La matrícula del estudiante debe estar activa en el sistema del IESAC. El estudiante no debe tener una restricción administrativa que impida su pre-inscripción.	
Garantía de Éxito:	El estudiante se pre-inscribió en línea.	



Escenario 2 Caso de uso 3 Consulta de Horarios

Nombre del Caso de uso:	Consulta de Horarios	
Actor(es):	Estudiante	
Descripción:	El estudiante podrá consultar en línea el horario de materias del cuatrimestre previamente pre-inscrito.	
Evento Desencadenador:	El estudiante inicia sesión con su matrícula y contraseña y dar clic en la opción "Pre-Inscripción" del menú de navegación.	
Pasos Realizados (ruta principal):	Información para los pasos:	
1. El estudiante inicia sesión en el sitio web ingresando su nombre de usuario y contraseña.	Nombre de usuario y contraseña	
2. El sistema muestra el menú de navegación y se selecciona la opción "Pre-Inscripción".	Menú de navegación	
3. Para mostrar el horario de materias del estudiante, debe haberse pre-inscrito con anterioridad, si ya fue pre-inscrito se mostrará el botón "Descargar Horario" dar clic sobre este botón.	Formulario de Consulta	
4. El sistema consulta el horario del estudiante en la base de datos acuerdo al grupo y cuatrimestre de este.	Conexión con la base de datos	
5. El horario de materias del estudiante se genera en formato PDF y se inicia el proceso de descarga. El horario tiene como columnas los días de la semana y como filas la hora en que se imparte la materia.	Librería Php para la generación de archivos en formato PDF	
Precondiciones:	El estudiante debe estar pre-inscrito a un cuatrimestre. El cuatrimestre debe tener un horario de materias relacionado con un grupo y carrera de la universidad.	
Garantía de Éxito:	El estudiante logró descargar el horario de materias del cuatrimestre pre-inscrito en formato PDF.	

4.1.3.4. Diagrama de Clases

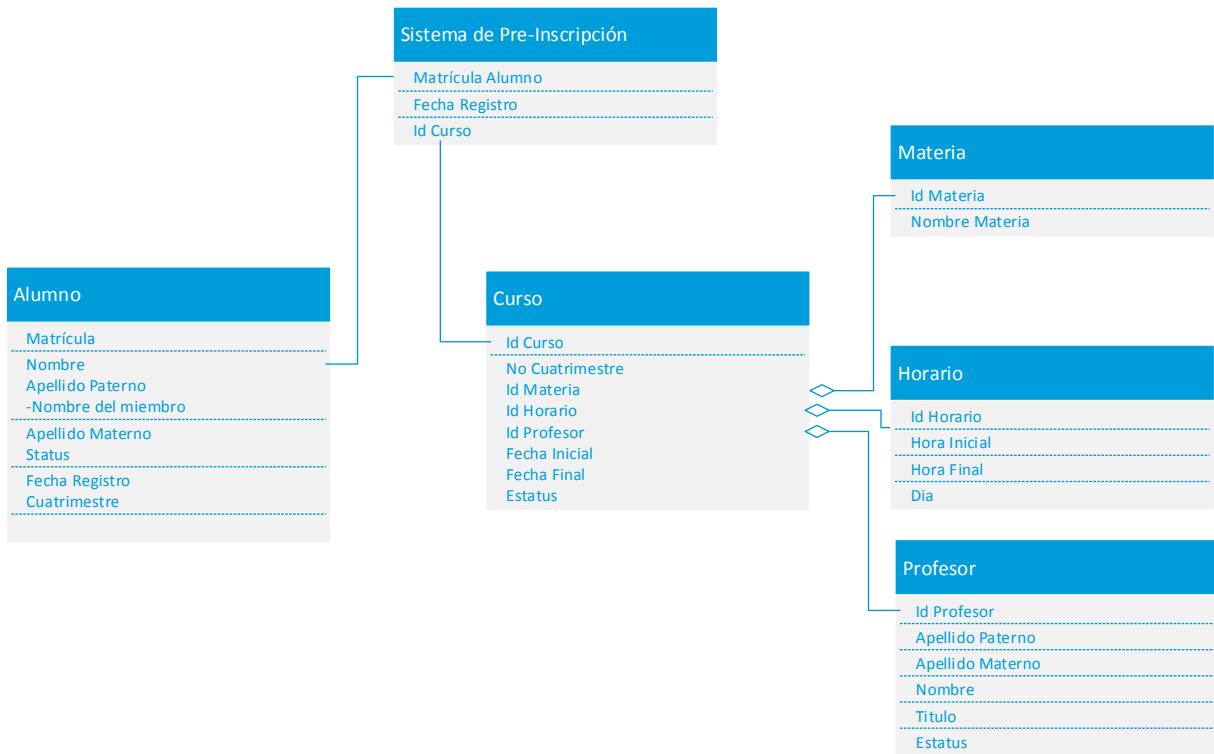


Figura 13: Diagrama de Clases

4.1.3.5. Diagramas de Secuencia

a) Coordinador de la Universidad

1) Diagrama de Secuencia de Registro de Cursos

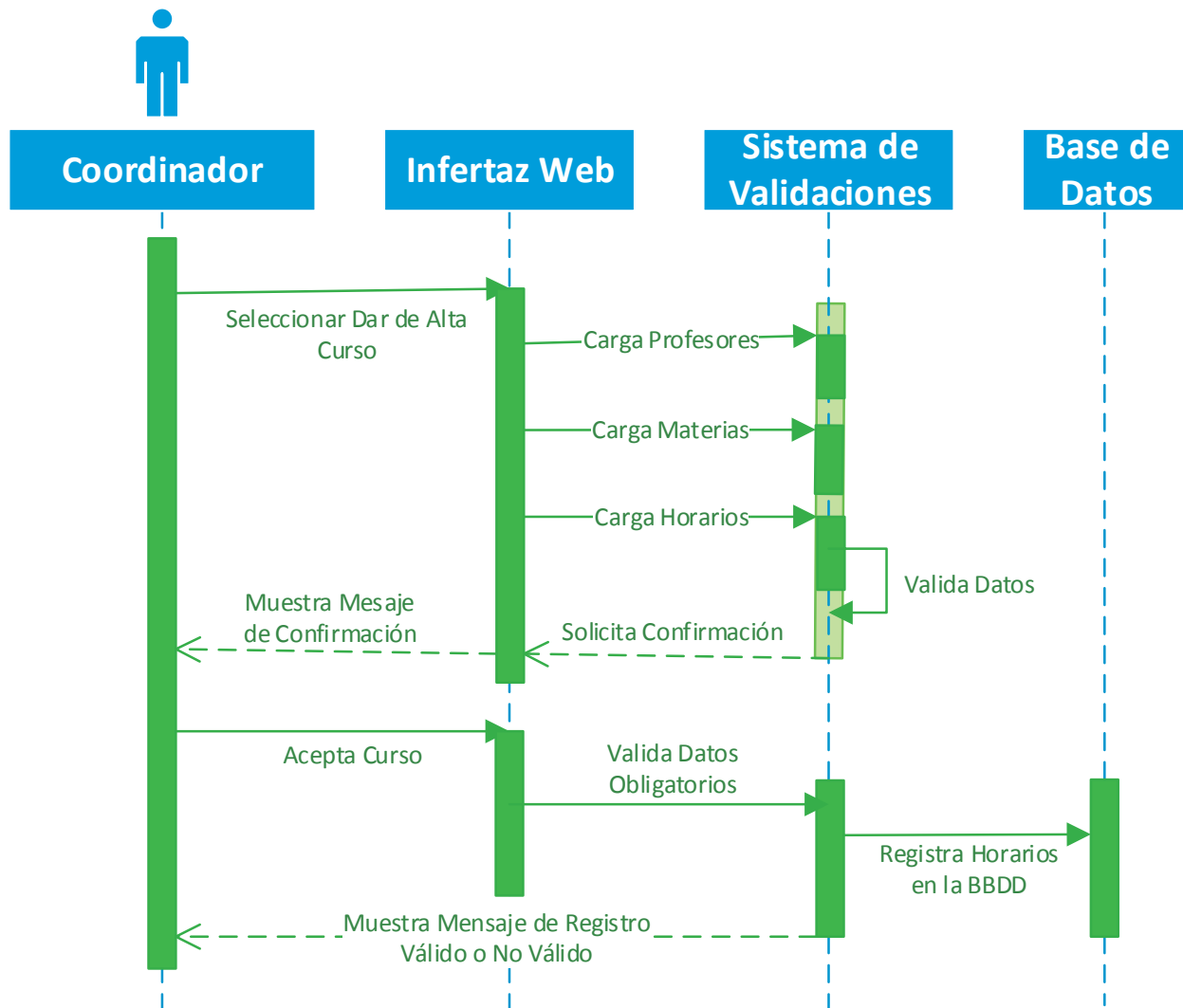


Figura 14: Diagrama de Secuencia de Registro de Cursos

2) Diagrama de Secuencia de Consulta de Horario

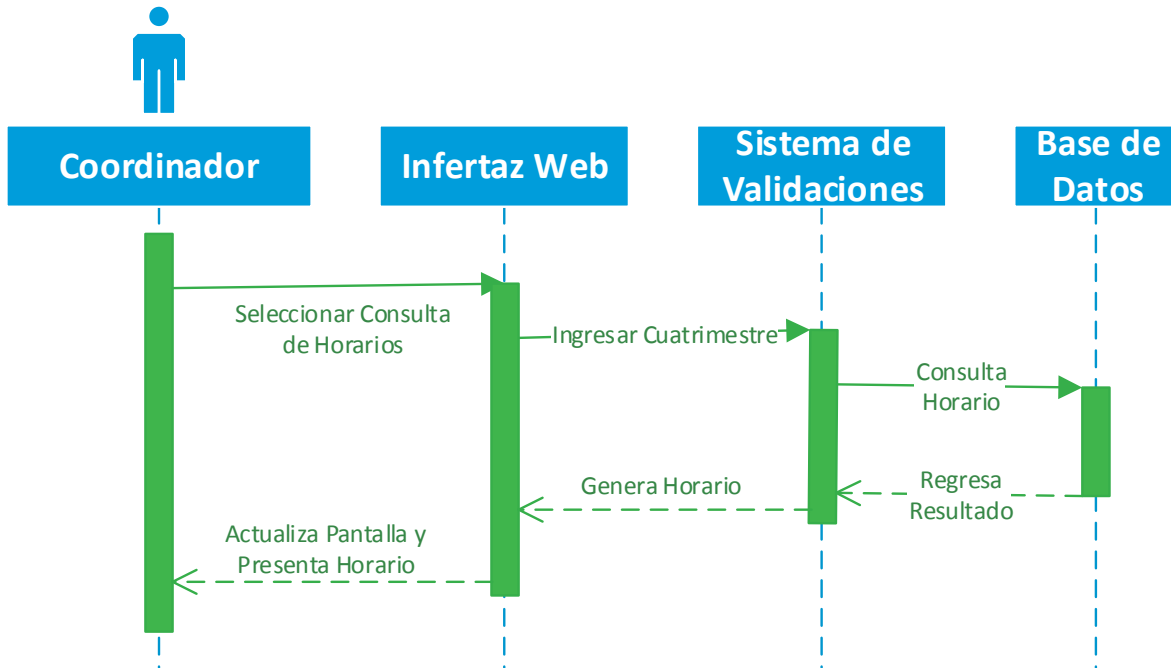


Figura 15: Diagrama de Secuencia de Consulta de Horario

3) Diagrama de Secuencia de Realización de Reportes

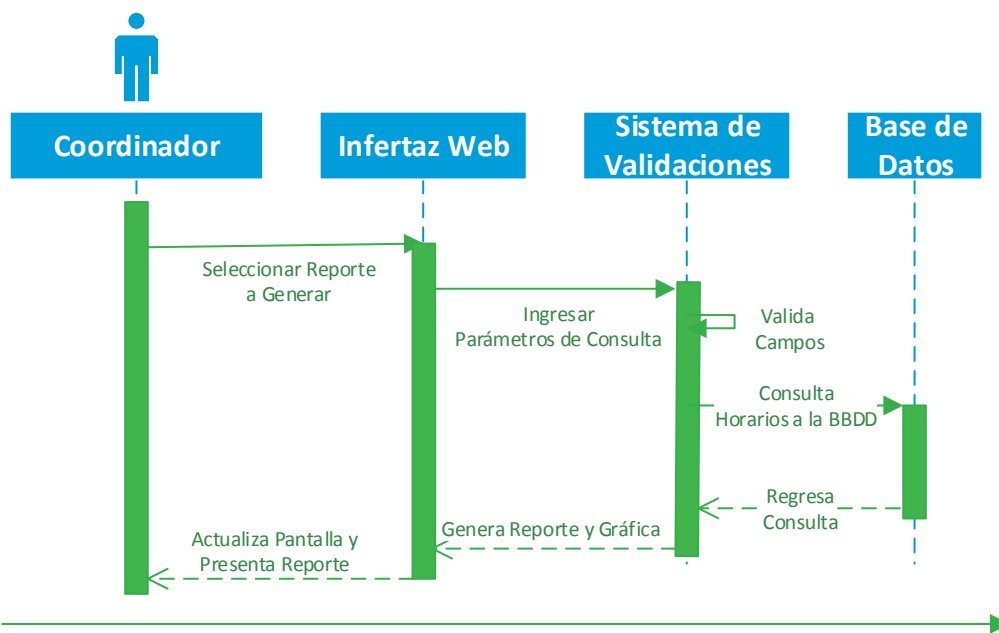


Figura 16: Diagrama de Secuencia de Realización de Reportes

b) Coordinador de la Universidad

1) Diagrama de Secuencia de Acceso al Sistema

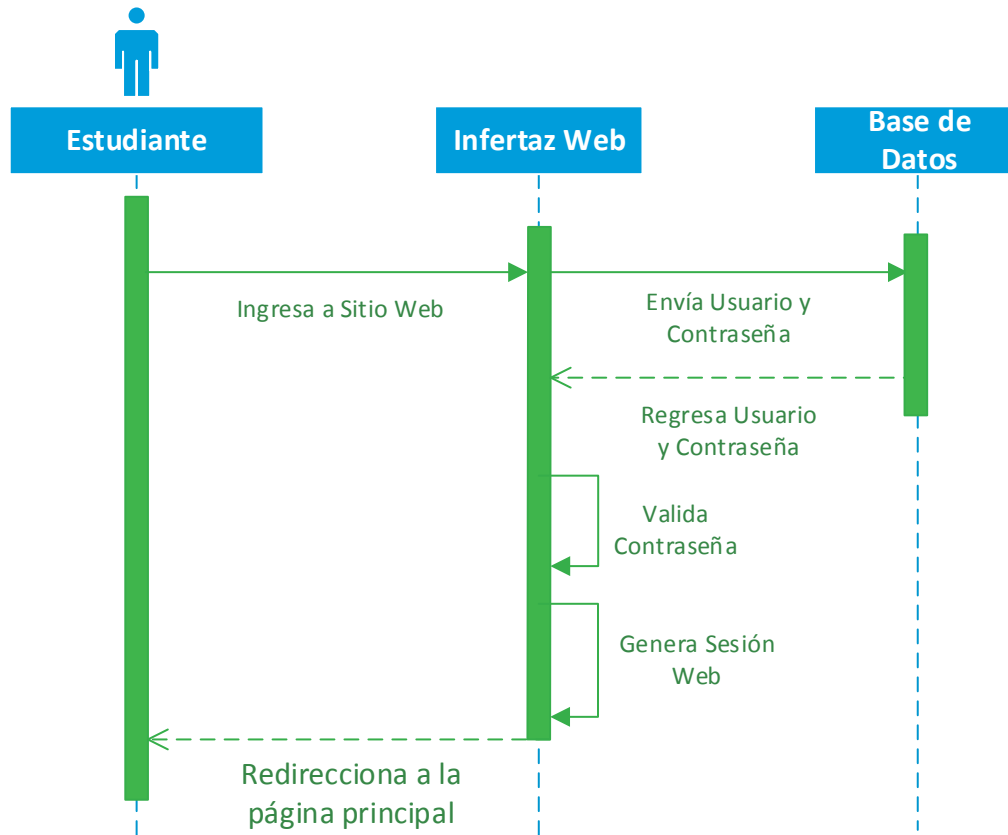


Figura 17: Diagrama de Secuencia de Acceso al Sistema

2) Diagrama de Secuencia de Pre-inscripción en línea

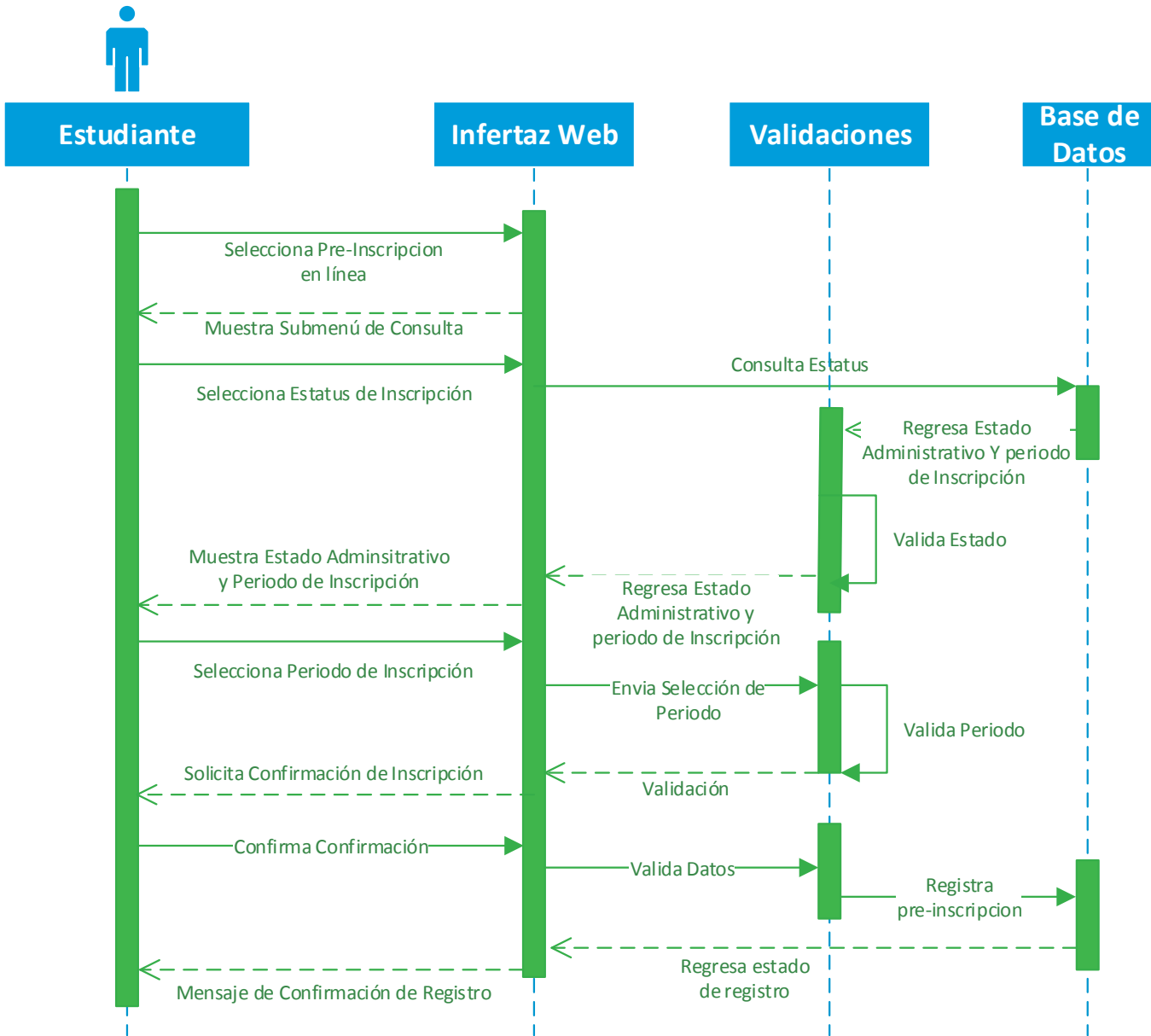


Figura 18: Diagrama de Secuencia de Pre-inscripción en línea

3) Diagrama de Secuencia de Consulta de Horario

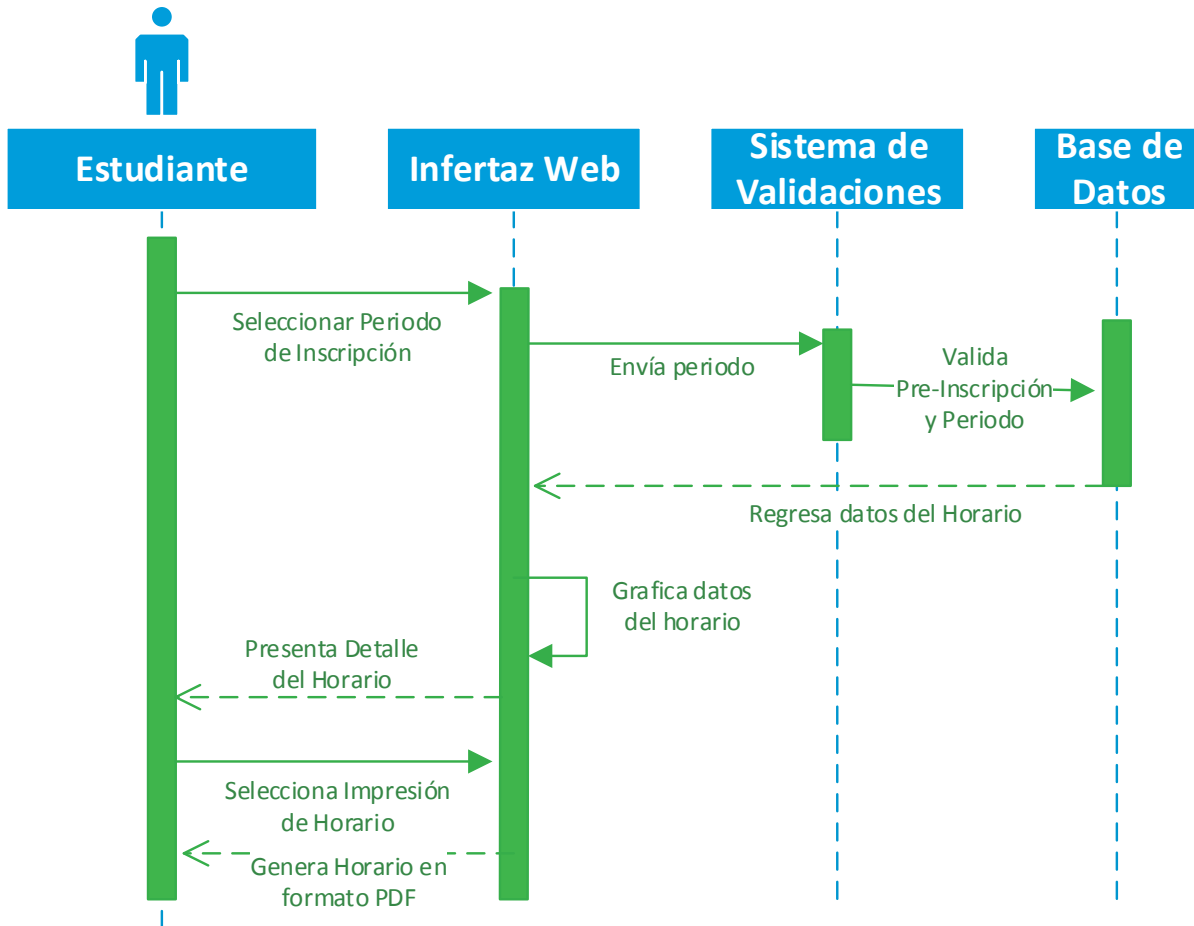


Figura 19: Diagrama de Secuencia de Consulta de Horario

5. Diseño del Sistema

5.1. Diseño de la Base de Datos

Después de realizar un análisis de toda la información obtenida con el Coordinador de la Universidad (usuario principal del sistema), se diseñó la Base de Datos cubriendo con todas las expectativas necesarias para un óptimo desempeño del sistema y con ello una mejor función para los usuarios.

5.1.1. Diagrama Físico

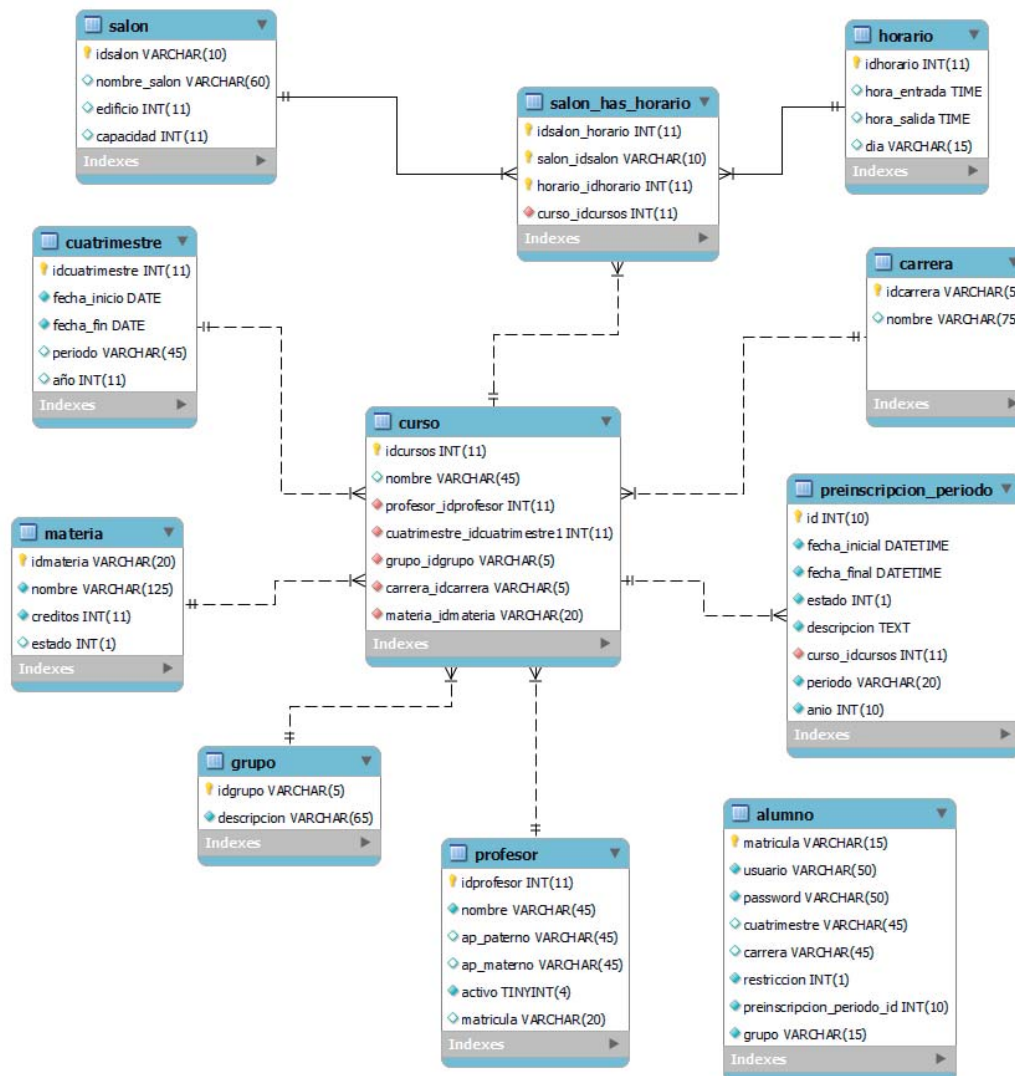


Figura 20 : Diagrama físico de la base de datos del sistema

5.1.2. Diagrama Entidad – Relación

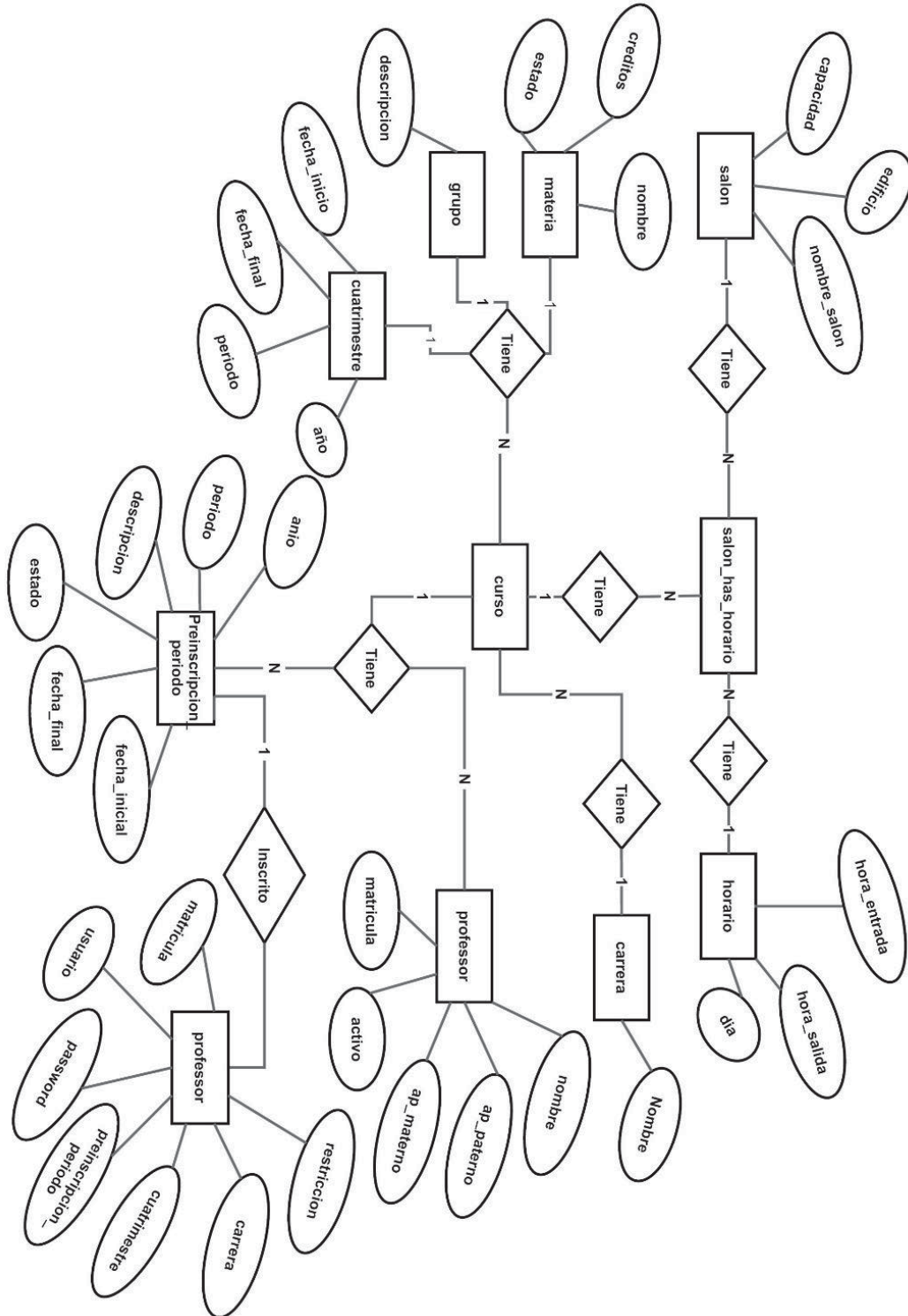


Figura 21: Diagrama de las relaciones entre las tablas del sistema.

5.2. Diseño de las Tablas de la Base de Datos

El sistema de pre-inscripción en línea de la Universidad IESAC utiliza las siguientes tablas para la administración de la información:

a) Tabla Alumno:

Contiene las columnas necesarias para identificar que los alumnos cuentan con un nombre de usuario, password para el acceso al sistema, así como la asignación de la carrera que cursa, grupo al que pertenece y si ya se encuentra preinscrito al cuatrimestre.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
matricula	VARCHAR(15)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
usuario	VARCHAR(50)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
password	VARCHAR(50)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
cuatrimestre	VARCHAR(45)			<input type="checkbox"/> BINARY	NULL
carrera	VARCHAR(45)			<input type="checkbox"/> BINARY	NULL
restriccion	INT(1)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0
preinscripcion...	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
grupo	VARCHAR(15)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL

Figura 22: Detalle de Tabla Alumno

b) Tabla Carrera:

Contiene la información relacionada con las carreras que se imparten en la universidad.

Column Name	Datatype	NOT NULL	AUTO INC	Flags
idcarrera	VARCHAR(5)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
nombre	VARCHAR(75)			<input type="checkbox"/> BINARY

Figura 23: Detalle de Tabla Carrera

c) Tabla Cuatrimestre:

Esta tabla se utiliza para llevar el control de los cuatrimestres, indicando el año y periodo que estará vigente.

Column Name	Datatype	NOT NULL	AUTO INC
idcuatrimestre	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
fecha_inicio	DATE	<input checked="" type="checkbox"/>	<input type="checkbox"/>
fecha_fin	DATE	<input checked="" type="checkbox"/>	<input type="checkbox"/>
periodo	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>
año	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>

Figura 24: Detalle de Tabla Cuatrimestre

d) Tabla Curso:

Contiene las columnas necesarias para registrar la información referente a las materias que se impartirán durante el cuatrimestre, ésta es la principal tabla de la base de datos ya que relaciona los horarios de materias, la carrera, el profesor que impartirá la materia, y el grupo de los estudiantes en base al cuatrimestre a cursar.

Column Name	Datatype	NOT NULL	AUTO INC
idcursos	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
nombre	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>
profesor_idpro...	INT(11)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cuatrimestre_i...	INT(11)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
grupo_idgrupo	VARCHAR(5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
carrera_idcarrera	VARCHAR(5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
materia_idmat...	VARCHAR(20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 25: Detalle de Tabla Curso

e) Tabla Grupo:

Contiene la información referente a los grupos que se crean por carrera y que determinan el cuatrimestre que cursa el estudiante.

Column Name	Datatype	NOT NULL	AUTO INC
idgrupo	VARCHAR(5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
descripcion	VARCHAR(65)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 26: Detalle de Tabla Grupo

f) Tabla Profesor:

Contiene la información necesaria para tener el registro de los profesores que imparten clases en la universidad.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
fecha_inicial	DATETIME	<input checked="" type="checkbox"/>			'2000-01-01 00:00:00'
fecha_final	DATETIME	<input checked="" type="checkbox"/>			'2000-01-01 00:00:00'
estado	INT(1)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0
descripcion	TEXT	<input checked="" type="checkbox"/>			NULL
curso_idcursos	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
periodo	VARCHAR(20)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
anio	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 27: Detalle de Tabla Profesor

g) Tabla Horario:

Esta tabla registra los horarios y los días de la semana que se impartirá una materia.

Column Name	Datatype	NOT NULL	AUTO INC
idhorario	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
hora_entrada	TIME		
hora_salida	TIME		
dia	VARCHAR(15)		

Figura 28: Detalle de Tabla Horario

h) Tabla Materia:

Esta tabla contiene la información de las materias impartidas en la universidad.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
idmateria	VARCHAR(20)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
nombre	VARCHAR(125)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	
creditos	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0
estado	INT(1)			<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 29: Detalle de Tabla Materia

i) Tabla Horario_temporal:

En esta tabla se registran altas, bajas y cambios de los horarios de las materias antes de ser registrados como oficiales.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
tag	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
materia	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
profesor	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
hora_entrada	TIME	<input checked="" type="checkbox"/>			'00:00:00'
hora_salida	TIME	<input checked="" type="checkbox"/>			'00:00:00'
dias	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
grupo	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
carrera	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL

Figura 30: Detalle de Tabla Horario_temporal

j) Tabla Preinscripcion_Periodo:

En esta tabla se registran los periodos de preinscripción en que el estudiante podrá pre-inscribirse.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
fecha_inicial	DATETIME	<input checked="" type="checkbox"/>			'2000-01-01 00:00:00'
fecha_final	DATETIME	<input checked="" type="checkbox"/>			'2000-01-01 00:00:00'
estado	INT(1)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0
descripcion	TEXT	<input checked="" type="checkbox"/>			NULL
curso_idcursos	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
periodo	VARCHAR(20)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
anio	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 31: Detalle de Tabla Preinscripcion_periodo

k) Tabla Salón:

Esta tabla dispone de la información necesaria para describir los salones de la universidad.

lumn Name	Datatype	NOT NULL	AUTO INC	Flags
idsalon	VARCHAR(10)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
nombre_salon	VARCHAR(60)			<input type="checkbox"/> BINARY
edificio	INT(11)			<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
capacidad	INT(11)			<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL

Figura 32: Detalle de Tabla Salón

l) Tabla Salón_has_horario:

Esta tabla mantiene la relación de los salones de la universidad con los horarios en el cual se impartirán las materias del cuatrimestre.

lumn Name	Datatype	NOT NULL	AUTO INC	Flags
idsalon_horario	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
salon_idsalon	VARCHAR(10)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
horario_idhora...	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
curso_idcursos	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL

Figura 33: Detalle de Tabla Salon_has_horario

5.3. Diseño de los Procedimientos Almacenados

Un procedimiento almacenado, es un conjunto de comandos SQL que se almacenan físicamente en una base de datos. Una vez que se hace, en respuesta a la petición de un usuario, es ejecutado directamente en el motor de la base de datos y como tal posee acceso directo a los datos que necesita manipular y solo necesita enviar los resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

El uso de los procedimientos almacenados incluye la validación de datos de entrada y en la encapsulación de procesos grandes y complejos, mejorando el rendimiento al manejar menos información entre el cliente y el servidor.

Los procedimientos almacenados son reutilizables, de manera de que los usuarios mediante la aplicación cliente no necesitan relanzar comando individuales, si no que pueden llamar el procedimiento para ejecutarlo en el servidor tantas veces como sea necesario.

A continuación se describen los procedimientos almacenados (stored procedure) de la base de datos del sistema:

a) Procedimiento: **USP_Login**

Este procedimiento es ejecutado en cada inicio de sesión en el sistema, validando los parámetros de entrada (usuario y contraseña) y posteriormente sea la validación de las credenciales de inicio de sesión en la base de datos del sistema,

```
DELIMITER $$
CREATE DEFINER='root'@'localhost' PROCEDURE `USP_LOGIN`(IN USUARIO VARCHAR(50), IN PASS VARCHAR(50))
BEGIN
    select a.usuario,a.password,if(c.idcoordinador is null, '0','1') as 'adminer' from alumno a
    left join coordinador c on a.matricula = c.matricula
    where a.usuario = USUARIO and a.password = md5(PASS);
END
```

Figura 34: Figura Procedimiento USP_LOGIN

b) Procedimiento: USP_PreInsCursos

Este procedimiento es ejecutado por el sistema para consultar si el estudiante ya se encuentra pre-inscrito a un cuatrimestre vigente, además de validar los parámetros de entrada (carrera y cuatrimestre).

```
DELIMITER $$  
  
CREATE DEFINER=`root`@`localhost` PROCEDURE `USP_PreInsCursos` (  
  in carrera varchar(45),  
  in cuatri int)  
BEGIN  
  select * from preinscripcion_periodo pp  
  inner join curso c  
    on pp.curso_idcursos = c.idcursos  
  and pp.estado = '1'  
  and c.carrera_idcarrera = carrera  
  and c.cuatrimestre_idcuatrimestre1 = cuatri;  
END
```

Figura 35: Procedimiento "USP_PreInsCursos"

c) Procedimiento: **USP_CursoQuery**

La consulta de los horarios de materias es frecuente en el sistema, esta requiere de información relacionada con maestros, salones, horarios y materias, por lo que se necesita la relación de múltiples tablas de la base de datos.

El procedimiento requiere como parámetro de entrada el id de la carrera y el id del cuatrimestre, obteniendo así la información necesaria para el procesamiento y su posterior distribución de esta en código HTML mediante Php.

```
DELIMITER $$  
  
CREATE DEFINER=`root`@`localhost` PROCEDURE `USP_CursoQuery`(in carrera varchar(25),  
in cuatri INT)  
BEGIN  
    SELECT car.idcarrera, car.nombre as 'NombreCarrera',  
        cuat.idcuatrimestre, cuat.fecha_inicio, cuat.fecha_fin, cuat.periodo, cuat.año,  
        h.dia, h.hora_entrada, h.hora_salida, h.idhorario,  
        mat.idmateria, mat.nombre as 'NombreMateria', concat(p.ap_paterno, ' ', p.ap_materno) as 'NombreProfesor'  
    FROM curso c  
    inner join carrera car      #CARRERA  
        on c.carrera_idcarrera = car.idcarrera  
    inner join materia mat      #MATERIA  
        on c.materia_idmateria = mat.idmateria  
    inner join cuatrimestre cuat #cuatrimestre  
        on c.cuatrimestre_idcuatrimestre1 = cuat.idcuatrimestre  
    inner join grupo gr        #grupo  
        on c.grupo_idgrupo = gr.idgrupo  
    inner join profesor p       #profesor  
        on c.profesor_idprofesor = p.idprofesor  
    inner join salon_has_horario shh  
        on c.idcursos = shh.curso_idcursos  
    inner join salon s          #salon  
        on shh.salon_idsalon = s.idsalon  
    inner join horario h  
        on shh.horario_idhorario = h.idhorario  
    where  
        c.carrera_idcarrera = carrera  
        and c.cuatrimestre_idcuatrimestre1 = cuatri;  
END
```

Figura 36: Procedimiento USP_CursoQuery

5.4. Diseño de Entradas

El sistema de pre-inscripción en línea demanda múltiples entradas, en esta sección trataremos las entradas que son requeridas por el sistema.

La entrada principal del sistema es el inicio de Sesión, donde se solicita al usuario (estudiante o coordinador) ingresar las credenciales de autenticación (usuario y contraseña) y la cadena de validación que despliega el reCaptcha (palabra de seguridad). La validación de las credenciales ingresadas, permite la autenticación del usuario y la presentación de los menús de navegación correspondientes.

The image shows a web form titled "Contact". It contains two text input fields: "Usuario*" and "Contraseña*". Below the password field is a reCAPTCHA widget with a red border. The widget includes a photo of a door with the number "152", a text input field with the placeholder "Type the text", and a "Privacy & Terms" link. At the bottom left of the form is a button labeled "Aceptar".

Figura 37: Formulario de Acceso al Sistema

Para realizar la carga de materias a un cuatrimestre, se requiere tener previamente el registro los siguientes elementos:

- Carreras
- Cuatrimestre
- Grupos
- Materias
- Profesores
- Salones.

El sistema permite el registro de todos los elementos mencionados anteriormente, en forma transparente para el usuario, mediante el uso de Ajax de JQuery, es decir: el usuario podrá cambiar de opciones de registro en una misma interfaz sin necesidad de cambiar entre varias páginas de navegación.

El menú con las opciones de registro se ilustra en la figura 38.



Figura 38: Menú de opciones de Registro

A continuación se describirán las entradas al sistema necesarias para registrar los horarios de materias de un cuatrimestre:

a) Carrera

Para el registro de una carrera que se impartirá en la universidad, es obligatorio introducir una clave única para su identificación y el nombre de la carrera. Estos campos son los mínimos necesarios para el sistema para correlacionarlos con el cuatrimestre y grupo de los estudiantes. La clave de la carrera puede ser alfanumérica. En la figura 39 se ilustra el formulario de registro de una carrera:

Un formulario con el título "Dar de Alta una Carrera" en negrita. El formulario contiene dos campos de texto: "Clave Carrera:" con un campo de entrada más corto, y "Titulo:" con un campo de entrada más largo. Debajo de los campos hay un botón rectangular con el texto "Aceptar".

Figura 39: Formulario de Registro de Carrera

b) Cuatrimestre

El formulario de registro de cuatrimestre, que se ilustra en la figura 40, permite el establecimiento de fechas que cubrirá el cuatrimestre, de igual manera el año y el periodo asignado de éste (primavera, verano u otoño).



Formulario de registro de Cuatrimestre. El formulario tiene un título "Dar de Alta un Cuatrimestre" en azul y negro. Contiene los siguientes campos:


- Fecha Inicio: 01-01-2014
- Fecha Final: 30-01-2014
- Periodo: Selecciona Periodo (menú desplegable)
- Año: (campo vacío)
- Botón Registrar

Figura 40: Formulario de Registro de Cuatrimestre

c) Grupo

Cada estudiante de la universidad debe estar asociado a un grupo, esta asociación permite darle el seguimiento a lo largo de la carrera. Cabe mencionar que cada cuatrimestre define la carga de materias que se impartirán.

Para el registro de grupos se requiere de una clave que servirá como identificador, ésta puede ser alfanumérica y opcionalmente podrá incluir una breve descripción. En la figura 41 se muestra el formulario de registro.



Formulario de registro de Grupo. El formulario tiene un título "Dar de Alta un Grupo" en azul y negro. Contiene los siguientes campos:

- Grupo Id: (campo vacío)
- Descripción: (campo vacío)
- Botón Registrar

Figura 41: Formulario de Registro de Grupo

d) Materia

En la figura 42 se ilustra el formulario de registro de las materias que son impartidas en la universidad, para ello se debe ingresar una clave única por materia que permitirá establecer la relación con el horario y el profesor que la impartirá. De igual manera se debe definir el nombre de la materia, el número de créditos y la asignación del estatus de la materia que puede ser activo o suspendido. El estatus de la materia puede variar según el caso, ya que pueden registrar materias pero que pueden estar suspendidas por diferentes razones, por ejemplo: falta de asignación del profesor que la impartirá etc.

Dar de Alta una Materia

Clave:

Nombre:

Créditos:

Estado:

Figura 42: Formulario de Registro de Materia

e) Profesor

El registro de profesores requiere de la siguiente información: nombre, apellidos, estatus (activo o no activo), y un número de matrícula, este último es un campo opcional, y debe ser llenado cuando el profesor fue alumno de la universidad. En la figura 43 se ilustra el formulario de registro:

Dar de Alta un Profesor

Nombre:

Apellido Paterno:

Apellido Materno:

Activo:

Matricula:

Figura 43: Formulario de Registro de Profesor

f) Salón

En la figura 44, se ilustra el formulario de registro para los salones, éste requiere de un id o abreviación única del salón, un nombre público para la identificación del salón, el edificio donde se encuentra y la capacidad, éste último campo es importante para evitar la sobre carga de estudiantes por salón



Dar de Alta un Salón

Salón Id:

Nombre:

Edificio:

Capacidad:

Figura 44: Formulario de Registro de Salón

g) Curso por Cuatrimestre

En la figura 45 se muestra el formulario principal, que permite relacionar las materias asignado un profesor que la impartirá, el horario y los días de la semana que se dará la clase, de la misma manera la asignación de grupos, carrera y cuatrimestre.

El formulario es 100% selectivo, lo cual permite seleccionar mediante listados los diferentes componentes de un horario, adicionalmente a esta característica tiene la posibilidad de adicionar o eliminar campos, haciendo eficiente la captura de los horarios.

Dar de Alta un Curso

Nombre: Carrera:

Grupo: Salón: Cuatrimestre: Año:

Materia	Profesor	Hora Inicio	Hora Fin	LU	MA	MI	JU	VI	SA	DO
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Materia	Profesor	Hora Inicio	Hora Fin	LU	MA	MI	JU	VI	SA	DO
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Registrar

Figura 45: Formulario de Registro de Cursos (Horario)

Otra entrada en el sistema es el registro de los periodos de pre-inscripción en línea, donde se registra el tipo de periodo (primavera, verano u otoño), el año, fecha y hora inicial y final que tendrá el estudiante para podrá ingresar al sistema en línea y consultar su estado académico, en caso de no contar una restricción podrá pre-inscribirse y consultar y/o descargar su horario de materias.

En la figura 46 se ilustra en formulario de registro y consulta de los periodos de preinscripción, esta opción solo está disponible para el coordinador de la universidad.

Cuando se realizar una consulta de los periodos registrados, el sistema presenta la información correspondiente en los campos, con posibilidad de modificarla, Por ejemplo: un motivo de actualización del periodo seria para extender la fecha u hora del acceso al sistema en línea que tendrá el estudiante para ingresar y poder pre-inscribirse.

Administrar Periodos de Pre-Inscripción

Periodo: Primavera 2014 [▼] [Buscar] [Nuevo Periodo]

Fecha - Hora Inicial: 01-01-2014 [00:00 ▼]

Fecha - Hora Final: 01-05-2014 [00:00 ▼]

Periodo: Primavera [▼]

Año: 2014

Estado: Activo [▼]

Descripción: Primavera 2014

[Actualizar Periodo]

Figura 46: Formulario de registro y consulta de los periodos de preinscripción

Una entrada al sistema es la pre-inscripción en línea del estudiante, si este no cuenta con una restricción académica el sistema mostrará un formulario simple de registro, donde solo aparecerá el periodo disponible para pre-inscribirse, como se muestra en la figura 47.

INICIO CONSULTAR ESTADO ACADÉMICO **PRE INSCRIPCIÓN** CONTACTO

Pre Inscripción en Línea

Selección de Periodo: Primavera 2014 [▼] [Aceptar]

Figura 47: Pre Inscripción en línea

El sistema de pre-inscripción en línea permite la consulta y modificación de los elementos de un horario (*carreras, cuatrimestre, grupos, materias, profesores, salones*), cuenta con formularios similares a los de registro.

El menú para la consulta y modificación de los elementos de un horario es igual al del registro. Ver figura 38 .

A continuación se describirán los formularios de actualización de cada uno de los elementos de los horarios de materias:

a) Carrera

El la figura 48 se ilustra el formulario de actualización de información de las carreras registradas en la universidad. Para la obtención de la búsqueda, el formulario cuenta con un listado de carreras para su selección y consulta, mostrando solo los campos modificables.

The image shows a web interface for updating career information. At the top, there is a label 'Seleccione' followed by a dropdown menu currently displaying 'Ciencias De La Computación'. To the right of the dropdown is a 'Consultar' button. Below this is a section titled 'Formulario de Actualización'. Inside this section, there is a label 'Titulo:' followed by a text input field containing 'Ciencias De La Computación'. At the bottom left of this section is an 'Aceptar' button.

Figura 48 Formulario de Actualización de Carreras

b) Cuatrimestre

En la figura 49, se muestra el formulario de consulta de los cuatrimestres registrados en la base de datos. El sistema solo habilitará para su modificación los campos que pueden ser editados.

Seleccione Cuatrimestre:2 Consultar

Formulario de Actualización

Fecha Inicio: 01-06-2014

Fecha Final: 01-08-2014

Periodo: Verano

Año: 2014

Aceptar

Figura 49: Formulario de Actualización de Cuatrimestres

c) Grupos

En la figura 50, se muestra el formulario para la consulta y actualización de la información de los Grupos registrados en la base de datos.

Seleccione Grupo:1A Consultar

Formulario de Actualización

Grupo Id: 1A

Descripción: Grupo I

Aceptar

Figura 50: Formulario de Actualización de Grupos

d) Materias

En la figura 51, se ilustra el formulario de consulta y actualización de la información relacionada con las materias.

Seleccione

Formulario de Actualización

Clave:

Nombre:

Créditos:

Estado:

Figura 51: Formulario de Actualización de Materias

e) Profesores

En la figura 52, se muestra el formulario de consulta y actualización de la información relacionada con los Profesores.

Seleccione

Formulario de Actualización

Nombre:

Apellido Paterno:

Apellido Materno:

Activo:

Matricula:

Figura 52: Formulario de Actualización de Profesores

f) Salones

En la figura 53, se muestra el formulario de consulta y actualización de la información relacionada con los Salones.

The image shows a web interface for updating hall information. At the top, there is a label 'Seleccione' followed by a dropdown menu showing 'Salón: Q2' and a 'Consultar' button. Below this is a section titled 'Formulario de Actualización' containing four input fields: 'Salón Id:' with the value 'Q2', 'Descripción:' with 'Sotano 2', 'Edificio:' with '104', and 'capacidad:' with '50'. An 'Aceptar' button is located at the bottom left of the form area.

Figura 53: Formulario de Actualización de Salones

5.5. Diseño de Salidas

La salida principal del sistema es la presentación de los horarios de clases del cuatrimestre según la carrera de la universidad consulta. Este módulo es consultado inicialmente por el coordinador de la universidad para revisar como quedaron organizadas las materias y posteriormente para la consulta de los estudiantes una vez que se haya pre-inscrito en línea a un cuatrimestre.

En la figura 54 se ilustra el formulario de consulta de horarios disponible para el coordinador de la universidad y la forma representativa de un horario de materias, adicionalmente a esto, el sistema ofrece la descarga de los horarios en formato PDF.

Consultar Horarios

CARRERA	Ciencias De La Computación ▼	CUATRIMESTRE	2014 - Primavera ▼	Aceptar
---------	------------------------------	--------------	--------------------	---------

Horario de Clases: Ciencias De La Computación Primavera 2014

HORA	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
07:00 hrs. a 08:59 hrs.	Calculo Aplicado (CLA1)	-	Calculo Aplicado (CLA1)	-	Calculo Aplicado (CLA1)	-
09:00 hrs. a 10:59 hrs.	-	Cultura y Etica Profesional I (Cul1)	Cultura y Etica Profesional I (Cul1)	Cultura y Etica Profesional I (Cul1)	-	-
11:00 hrs. a 12:59 hrs.	-	-	-	-	-	-
13:00 hrs. a 14:59 hrs.	-	-	-	-	-	-
15:00 hrs. a 16:59 hrs.	-	-	-	-	-	-
17:00 hrs. a 18:59 hrs.	-	Graficación (Gra1)	Graficación (Gra1)	-	Graficación (Gra1)	-
19:00 hrs. a 20:59 hrs.	-	-	-	-	-	-

Descargar Horario

Figura 54: Horario de Clases

Otra salida del Sistema es el módulo de consulta de restricciones académicas, que le permite a los estudiantes consultar si tiene alguna restricción que impida su pre-inscripción en línea, por ejemplo: adeudo de una colegiatura, etc.

En la figura 55 se muestra la forma en que el sistema informa al estudiante su estado académico, en este caso el estudiante no podrá pre-inscribirse en línea.

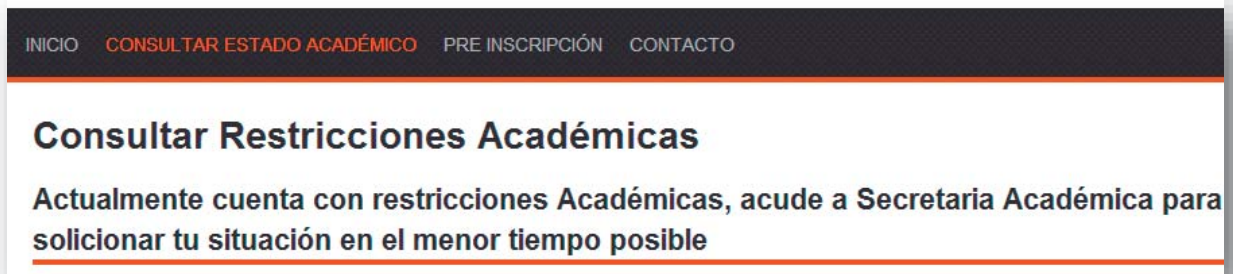


Figura 55: Estado Académico 1

Si el estudiante no presentara una restricción académica, el sistema desplegará una notificación como se muestra en la figura 56.



Figura 56: Estado Académico 2



6. Conclusiones y Trabajos Futuros

6.1. Conclusiones

El Sistema de pre-inscripción en línea de la Universidad IESAC permite a los estudiantes pres-inscribirse de una manera más automatizada y por ende la agilización en los procesos de la información para los usuarios (estudiantes y coordinador de la universidad).

Con el desarrollo del sistema se aprendieron muchas cosas, entre las que destaca el uso de sistemas de seguridad (reCaptchas) tanto para la validación de las credenciales del usuario antes de ingresar al sistema como en el manejo de sentencias preparadas en el DBMS mediante PDO. Con ellos podemos combinar ambas tecnologías que están al alcance de nuestras manos para desarrollar software con excelente manufactura, seguridad y eficiencia.

En esta versión del sistema de pre-inscripción, tiene sus ventajas y desventajas como toda aplicación de software.

Algunas ventajas:

- Sencillez en el registro de los horarios de materias de las diferentes carreras de la universidad.
- Agilización en el proceso de pre-inscripción mediante su automatización.
- Disponibilidad y facilidad para la consulta de los horarios por los estudiantes.
- Consulta y generación de reportes en forma remota mediante el sitio Web.

Entre las limitantes del sistema podemos observar las siguientes:

- Tiempo de vida del Software y Hardware.
- Seguridad en la página web, es decir, agregar certificados de seguridad SSL y encriptación de datos.
- Avance en las tecnologías empleadas.



Con todo ello podemos decir que ha sido de gran experiencia poder realizar un sistema para una universidad, por la demanda y el uso que tendrá éste, permitiéndome el contacto con nuevas tecnologías principalmente las de seguridad tanto en nivel Web como en las transacciones con la base de datos.

6.2. Trabajos Futuros

La primera línea de continuación con este trabajo es el desarrollo de una versión más eficiente a la hora de generar los horarios, incluyendo reglas de validación, por ejemplo: el sistema evitará que un profesor tenga una carga de trabajo más grande de su capacidad, el sistema sugerirá a profesores son candidatos para impartir una materia al momento de asignar la materias y evitar el traslape de materias en los horarios por salón.

A corto plazo se piensa la posibilidad de cambiar el método de consulta de la información del estudiante, mediante la consulta al WebServices del sistema actual de la universidad Sunny Escolar y de esta manera generar el envío automático de notificaciones por correo electrónico al estudiante informado los periodos de pre-inscripción de la universidad.

La generación de Códigos QR en la página web, a raíz del incremento del uso de Smartphone's en la comunidad estudiantil, que en la actualidad mediante App (aplicaciones móviles) facilita la lectura de estos códigos, para que el usuario pueda descargar de manera automática a su móvil el calendario de clases al momento de leer el código QR.

Finalmente se propone el uso de certificados de seguridad SSL, que son los encargados de cifrar toda la información que viaja entre las computadoras y la página web, para evitar el robo de datos y daños a la reputación de la universidad.



Índice de Figuras

FIGURA 1: SIMBOLOGÍA DEL MODELO ENTIDAD RELACIÓN	10
FIGURA 2: MODELO CLIENTE - SERVIDOR.....	21
FIGURA 3: ESQUEMA DEL FUNCIONAMIENTO DE LAS PÁGINAS PHP.....	25
FIGURA 4 : MODELO LINEAL SECUENCIAL (MODELO EN CASCADA)	27
FIGURA 5: CLASE.....	32
FIGURA 6 REPRESENTACIÓN VISUAL DE UNA GENERALIZACIÓN EN UML	33
FIGURA 7 REPRESENTACIÓN VISUAL DE UNA ASOCIACIÓN EN UML	34
FIGURA 8 REPRESENTACIÓN VISUAL DE UNA RELACIÓN DE ACUMULACIÓN EN UML	35
FIGURA 9 REPRESENTACIÓN VISUAL DE UNA RELACIÓN DE ACUMULACIÓN EN UML	35
FIGURA 10: DIAGRAMA DE FLUJO DE DATOS A NIVEL CONTEXTO	38
FIGURA 11: CASO DE USO COORDINADOR DE LA UNIVERSIDAD	39
FIGURA 12: CASO DE USO ESTUDIANTE.....	40
FIGURA 13: DIAGRAMA DE CLASES	49
FIGURA 14: DIAGRAMA DE SECUENCIA DE REGISTRO DE CURSOS.....	50
FIGURA 15: DIAGRAMA DE SECUENCIA DE CONSULTA DE HORARIO.....	51
FIGURA 16: DIAGRAMA DE SECUENCIA DE REALIZACIÓN DE REPORTES.....	51
FIGURA 17: DIAGRAMA DE SECUENCIA DE ACCESO AL SISTEMA.....	52
FIGURA 18: DIAGRAMA DE SECUENCIA DE PRE-INSCRIPCIÓN EN LÍNEA	53
FIGURA 19: DIAGRAMA DE SECUENCIA DE CONSULTA DE HORARIO.....	54
FIGURA 20 : DIAGRAMA FÍSICO DE LA BASE DE DATOS DEL SISTEMA	55
FIGURA 21: DIAGRAMA DE LAS RELACIONES ENTRE LAS TABLAS DEL SISTEMA.....	56
FIGURA 22: DETALLE DE TABLA ALUMNO.....	57
FIGURA 23: DETALLE DE TABLA CARRERA.....	57
FIGURA 24: DETALLE DE TABLA CUATRIMESTRE	58
FIGURA 25: DETALLE DE TABLA CURSO.....	58
FIGURA 26: DETALLE DE TABLA GRUPO	58
FIGURA 27: DETALLE DE TABLA PROFESOR.....	59
FIGURA 28: DETALLE DE TABLA HORARIO	59
FIGURA 29: DETALLE DE TABLA MATERIA	59
FIGURA 30: DETALLE DE TABLA HORARIO_TEMPORAL	60
FIGURA 31: DETALLE DE TABLA PREINSCRIPCION_PERIODO.....	60
FIGURA 32: DETALLE DE TABLA SALÓN.....	61
FIGURA 33: DETALLE DE TABLA SALON_HAS_HORARIO	61
FIGURA 34: FIGURA PROCEDIMIENTO USP_LOGIN	62
FIGURA 35: PROCEDIMIENTO "USP_PREINSCURSOS	63
FIGURA 36: PROCEDIMIENTO USP_CURSOQUERY	64
FIGURA 37: FORMULARIO DE ACCESO AL SISTEMA	65
FIGURA 38: MENÚ DE OPCIONES DE REGISTRO.....	66
FIGURA 39: FORMULARIO DE REGISTRO DE CARRERA.....	66
FIGURA 40: FORMULARIO DE REGISTRO DE CUATRIMESTRE	67
FIGURA 41: FORMULARIO DE REGISTRO DE GRUPO.....	67
FIGURA 42: FORMULARIO DE REGISTRO DE MATERIA	68
FIGURA 43: FORMULARIO DE REGISTRO DE PROFESOR.....	68
FIGURA 44: FORMULARIO DE REGISTRO DE SALÓN	69
FIGURA 45: FORMULARIO DE REGISTRO DE CURSOS (HORARIO).....	70
FIGURA 46: FORMULARIO DE REGISTRO Y CONSULTA.....	71



FIGURA 47: PRE INSCRIPCIÓN EN LÍNEA	71
FIGURA 48: FORMULARIO DE ACTUALIZACIÓN DE CARRERAS	72
FIGURA 49: FORMULARIO DE ACTUALIZACIÓN DE CUATRIMESTRES	73
FIGURA 50: FORMULARIO DE ACTUALIZACIÓN DE GRUPOS	73
FIGURA 51: FORMULARIO DE ACTUALIZACIÓN DE MATERIAS	74
FIGURA 52: FORMULARIO DE ACTUALIZACIÓN DE PROFESORES	74
FIGURA 53: FORMULARIO DE ACTUALIZACIÓN DE SALONES	75
FIGURA 54: HORARIO DE CLASES.....	76
FIGURA 55: ESTADO ACADÉMICO 1	77
FIGURA 56: ESTADO ACADÉMICO 2	77



Bibliografía.

- Introducción a los Sistemas de Bases de Datos C.J Date (7a Edición),
Pearson Prentice Hall
2001
- Análisis y Diseño de Sistemas de Información Jeffrey L. Whitten / Lonnie D. Bentley /
Victor M. Barlow
M.C. Grew Hill
2003
- Ingeniería de Software Ian Sommerville (7a Edición),
Pearson Prentice Hall
2005
- Fundamentos de Php Vikram Vaswani
M.C. Grew Hill
2009
- Fundamentos de SQL Andy Opper / Robert Sheldon, (3a
Edición)
M.C. Grew Hill
2009
- Javascript Jim Keogh
Anaya Multimedia
2005, España
- Fundamentos de jQuery Rebecca Murphey, Autoedición
2011
- <http://www.w3.org/>
- <http://www.apache.org/>
- http://es.wikipedia.org/wiki/PHP_Data_Objects
- http://librosweb.es/javascript/capitulo_1.html

