UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"Consulta de Información de Estatus del Personal Académico y Periodos de Examen por Semestre de la DIE"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

GUTIÉRREZ VELÁZQUEZ GABRIELA OFELIA

RODRÍGUEZ NEGRETE MARICELA VERÓNICA

DIRECTOR DE TESIS: ING. MARICELA CASTAÑEDA PERDOMO

MÉXICO, D. F. 2008

AGRADECIMIENTOS

A mis padres mil gracias por estar siempre a mi lado brindándome su apoyo y amor incondicional, para poder alcanzar esta meta que también es suya.

Y a ti mi amor por estar siempre a mi lado y darme todo tu amor, confianza y la fuerza para seguir siempre adelante.

Gabriela Ofelia Gutiérrez Velázquez

Gracias a Dios, por permitirme terminar esta etapa.

A la memoria de mi padre, por ser un gran ejemplo en mi vida, por ser el motor fundamental para terminar mi formación profesional y ahora ser un ángel para mi.

A mi madre por su amor incondicional, apoyo y fortaleza, por darme el más valioso legado que es mi educación, le dedico este trabajo y mis futuros triunfos.

A mi esposo, por darme aliento para seguir a pesar de las adversidades, enseñarme que la sencillez es más valiosa que cualquier cosa y ser mi fuente de amor e inspiración .

A mi hermana Ana porque fue mi punto de admiración desde niña.
A mi hermana Elsa por sus consejos y cariño.
A mi cuñado por ser un amigo y mi hermano.

A mi grupo de trabajo en la Facultad de Ingeniería por enseñarme la esencia de ser ingeniero y amigo.
A mis amigos que a pesar de las distancias, como muestra de agradecimiento por su apoyo y tantos momentos imborrables en mi vida.

A mis suegros que han demostrado ser grandes personas, por sus valores y su generosidad.

Ing. Gloria Martínez Rosas, por darme el último empujón para terminar.

Así como las personas que formaron parte de este trabajo y que ya no pudieron estar, tienen su lugar reservado desde el cielo.

Maricela Verónica Rodríguez Negrete

“Felicidad no es solamente sumar tus conquistas, sino soñar con aquellas que aún tienes por realizar”. (Anónimo)

ÍNDICE

INTRODUCCIÓN

Introducción

En la División de Ingeniería Eléctrica (DIE), se necesita un sistema de consulta que ayude al personal académico a mantenerse informado de su situación laboral dentro de la Facultad de Ingeniería y que pueda consultar los periodos de exámenes extraordinarios.

En el primer capítulo hablaremos con más detalle de la situación actual de la DIE con su personal académico, la importancia de un sistema de consulta para mejorar la comunicación entre ambos, así como los requerimientos del mismo.

El segundo capítulo se refiere a las bases teóricas necesarias para el desarrollo del sistema, además se realiza una comparación con algunas herramientas donde se verán las características, ventajas y desventajas de estas, y así llegar a la solución óptima.

En el tercer capítulo se da a conocer el ciclo de vida de un sistema y en el último capítulo se llevarán a cabo las pruebas.

Capítulo 1

Situación Actual

Capítulo 1. Situación Actual

1.1. Definición y acotamiento del problema.

La Facultad de Ingeniería de la UNAM, esta dividida en 7 divisiones para una mejor organización y desempeño de sus funciones; en esta ocasión se hace referencia a la División de Ingeniería Eléctrica (DIE). En la figura 1 podemos observar su organización.

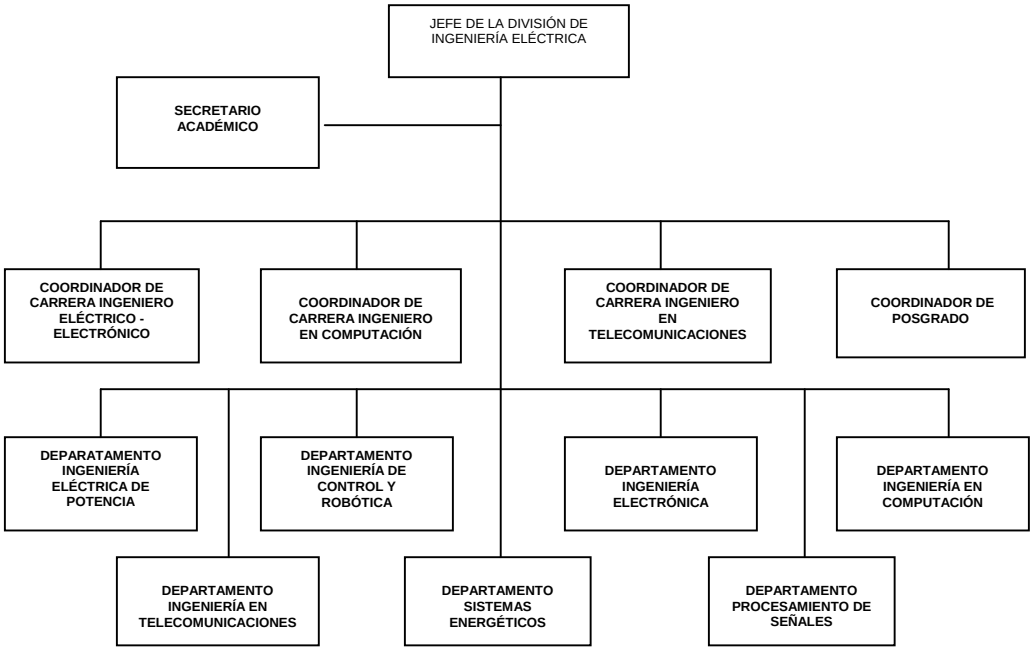


Figura 1. Organigrama de la División de Ingeniería Eléctrica.

La Secretaria Académica se apoya en la Secretaria Auxiliar en todo lo relacionado al personal académico.

En la actualidad la división, cuenta con un sitio WEB donde se puede obtener información general, de las carreras, se puede acceder a páginas de alumnos y profesores, puede realizar visitas virtuales dentro por la división, tiene ligas que conectan con los distintos departamentos, laboratorios, sociedades de alumnos y con la página principal de la facultad, así como algunos servicios por ejemplo: correo para profesores y alumnos, consulta de libros digitales, etcétera.

Sin embargo carece de un sistema donde el personal académico de esta división, pueda consultar su estatus y los periodos de exámenes por semestre. Por lo que éste tiene que ir o llamar a la oficina de la secretaria auxiliar o al departamento correspondiente para obtener dicha información, esto genera una pérdida de tiempo, esfuerzo y recursos innecesaria, ya que la división cuenta con los elementos suficientes para hacerlo, por un medio electrónico.

1.2 Requerimientos del sistema

La secretaría cuenta con un sistema programado en Visual Basic, en el cual se encuentra la información del estatus del personal académico y el uso, es para su personal. Aunque se puede contar con una copia de la base de datos generada por este sistema, con el fin de que nos proporcione la información requerida, dicha base fue hecha en Access. Por otro lado, servicios escolares proporciona los periodos de exámenes al comienzo de cada semestre.

Como ya se había mencionado en el punto anterior, se tiene un espacio en la red que está disponible para la comunidad y el cual podemos aprovechar.

Por esta razón decidimos crear un sistema de consulta, donde se tenga el acceso a la base de datos del personal académico de la división, y que este a su vez cuente con una interfaz amigable para el usuario a través de la red, así como para el personal que maneje el sistema para su administración.

Capítulo 2

Herramientas de desarrollo

Capítulo 2. Herramientas de desarrollo

Para el desarrollo del sistema, necesitamos de un lenguaje de programación y un manejador de bases de datos, a continuación, se hablará de algunos de ellos y se hará una comparación para escoger el más adecuado.

2.1. Teoría básica de bases de datos.

Para entender de una manera sencilla lo que es una base de datos daremos algunas definiciones que la involucran.

- **Dato.** Es un conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos.
- **Información.** Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario. Para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos.
- **Campo.** Es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto.
- **Registro.** Es una colección de campos de iguales o de diferentes tipos.
- **Archivo.** Es una colección de registros almacenados siguiendo una estructura homogénea.
- **Base de Datos.** Es una colección de archivos interrelacionados, son creados con un DBMS. El contenido de una base de datos engloba a la información concerniente (almacenadas en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.
- **Sistema Manejador de Base de Datos (DBMS.)** Es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. Hablaremos más adelante todo lo concerniente a un DMBS.
- **Instancia**. Se refiere al estado que presenta una base de datos en un tiempo dado.
- **Esquema de base de datos.** Es la estructura por la que esta formada la base de datos, se especifica por medio de un conjunto de definiciones que se expresa mediante un lenguaje especial llamado lenguaje de definición de datos. (DDL).
- **Lenguaje de definición de datos.** Denominado por sus siglas como: DDL (Data Definition Language). Permite definir un esquema de base de datos por medio de una serie de definiciones que se expresan en un lenguaje especial, el resultado de estas definiciones se almacena en un archivo especial llamado diccionario de datos.
- **Lenguaje de manipulación de datos.** Se refiere a las operaciones de insertar, recuperar, eliminar o modificar datos; dichas operaciones son realizadas a través del lenguaje de manipulación de datos (DML, Data Manipulation Language), que es quién permite el acceso de los usuarios a los datos.

Existen básicamente 2 tipos de lenguajes de manipulación de datos:

- Procedimentales:
  Los DML requieren que el usuario especifique que datos se necesitan y cómo obtenerlos.
- No procedimentales:
  Los DML requieren que el usuario especifique que datos se necesitan y sin especificar cómo obtenerlos.

2.1.1 Sistema Manejador de Base de Datos

Los sistemas de base de datos se diseñan para manejar grandes cantidades de información, la manipulación de los datos involucran tanto la definición de estructuras para el almacenamiento de la información como la provisión de mecanismos para la manipulación de la información, además un sistema de base de datos debe de tener implementados mecanismos de seguridad que garanticen la integridad de la información, a pesar de caídas del sistema o intentos de accesos no autorizados.

Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la base de datos, en la figura 2 se puede ver de manera mas clara lo anterior.

El DBMS es conocido también como Gestor de Base de datos.

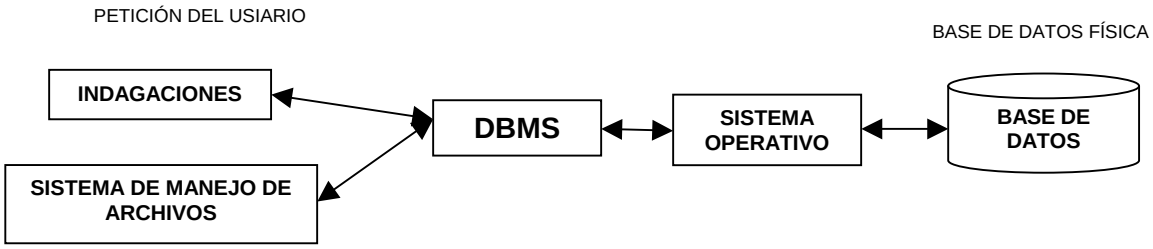


Figura 2. Interfase entre la base de datos física y las peticiones del usuario

El DBMS interpreta las peticiones de entrada / salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

Los propósitos principales de un sistema de base de datos es disminuir los siguientes aspectos:

- Redundancia e inconsistencia de datos**.**

  Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea más de una vez en un dispositivo de almacenamiento. Esto aumenta los costos de almacenamiento y acceso a los datos, además de que puede originar la inconsistencia de los datos, es decir, diversas copias de un mismo dato no concuerdan entre si, por ejemplo: se actualiza la dirección de un cliente en un archivo y que en otros archivos permanezca la anterior.

- 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. Supóngase un banco, y que uno de los gerentes necesita averiguar los nombres de todos los clientes que viven dentro del código postal 78733 de la ciudad. El gerente pide al departamento de procesamiento de datos que genere la lista correspondiente. Puesto que esta situación no fue prevista en el diseño del sistema, no existe ninguna aplicación de consulta que permita este tipo de solicitud, esto ocasiona una deficiencia del sistema.

- 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.

- Anomalías del acceso concurrente.

  Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente. En un entorno así la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

- 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 autentificación y protección de los datos. En un banco por ejemplo, el personal de nóminas sólo necesita ver la parte de la base de datos que tiene información acerca de los distintos empleados del banco y no a otro tipo de información.

- 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.

Un propósito importante de un sistema manejador de base de datos es proporcionar a los usuarios una visión *abstracta* de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente.

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema; Interno, conceptual y externo, específicamente el de almacenamiento físico, el del usuario y el del programador.

- Nivel físico.

  Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en como se almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).

- Nivel conceptual.

  El siguiente nivel más alto de abstracción, describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.

El nivel conceptual consta de las siguientes definiciones:

1. Definición de los datos: Se describen el tipo de datos y la longitud de campo todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
2. Relaciones entre datos*:* Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

✶ Nivel de visión.

Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario acreditado para verla. El sistema puede proporcionar muchas visiones para la misma base de datos.

La interrelación entre estos tres niveles de abstracción se ilustra en la figura 3.

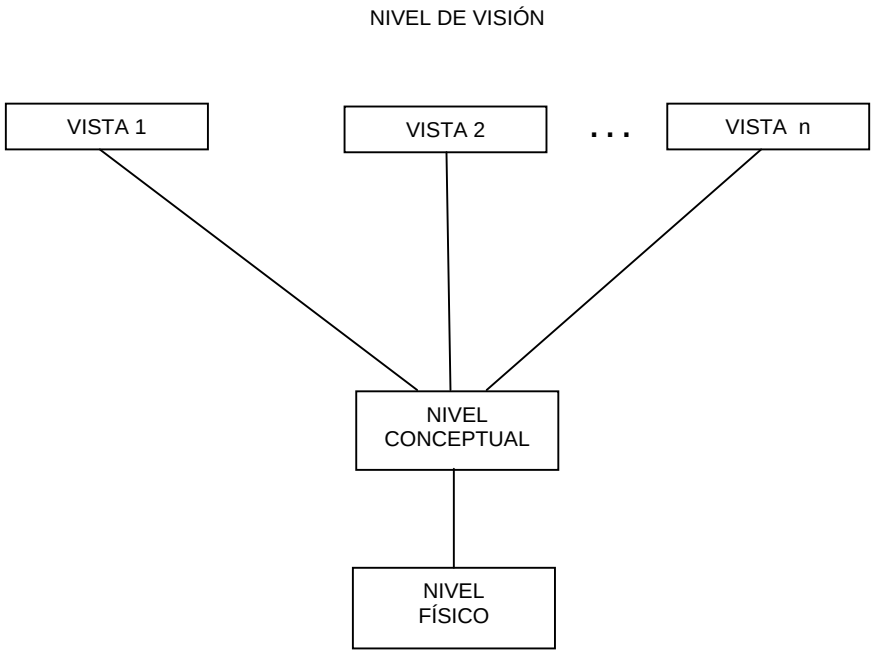


Figura 3. Niveles de abstracción de la información en una base de datos

2.1.2 Modelo de Datos

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos:

1. Modelos lógicos basados en objetos.

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

- Modelo Entidad-Relación.

Denominado por sus siglas como: E-R: este modelo representa a la realidad a través de *entidades***,** que son objetos que existen y que se distinguen de otros por sus características, por ejemplo: un alumno se distingue de otro por sus características particulares como lo es el nombre, o el numero de control asignado al entrar a una institución educativa, así mismo, un empleado, una materia, etc. Las entidades pueden ser de dos tipos:

- Tangibles: Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

- Intangibles: Todos aquellos eventos u objetos conceptuales que no podemos ver, aún sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

Las características de las entidades en base de datos se llaman *atributos***,** por ejemplo el nombre, dirección, teléfono, grado, grupo, etc. son atributos de la entidad alumno; Clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de *relaciones.*

Bueno, ahora nos falta describir como se representa un modelo E-R gráficamente, la representación es muy sencilla, se emplean símbolos, los cuales se representan en la figura 4.

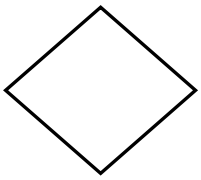
Símbolo	Representa
	ENTIDAD
	RELACIÓN
	ATRIBUTOS
	LIGAS

Figura 4. Símbolos de representación de E-R

2. Modelos lógicos basados en registros.

Se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.

Los tres modelos de datos más ampliamente aceptados son:

✶ Modelo Relacional

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características(atributos) de cada registro localizado en la tupla;

Existen dos formas de representarla; pero para ello necesitamos definir que es una **llave primaria:** Es un atributo el cual definimos como atributo principal, es una forma única de identificar a una entidad. Por ejemplo, el RFC de un empleado se distingue de otro por que los RFC no pueden ser iguales.

Ahora si, las formas de representar las relaciones en este modelo son:

1. Haciendo una tabla que contenga cada una de las llaves primarias de las entidades involucradas en la relación.
2. Incluyendo en alguna de las tablas de las entidades involucradas, la llave de la otra tabla.

- Modelo de Red

  Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

- Modelo Jerárquico

  Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

3. Modelos físicos de datos.

   Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos. Existen dos clasificaciones de este tipo que son:

   - Modelo unificador.
   - Memoria de elementos.

2.1.3 Independencia de los Datos

Se refiere a la protección contra los programas de aplicación que puedan originar modificaciones cuando se altera la organización física o lógica de la base de datos. Existen 2 niveles de independencia de datos.

- **Independencia física de datos:** Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación.
- **Independencia lógica de datos:** Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.

2.2. Características, ventajas y desventajas del manejador de bases de datos

2.2.1 SQL

Structured Query Language no es más que un lenguaje estándar de comunicación con bases de datos, el cual nos permite realizar tablas y obtener datos de ella de manera muy sencilla.

Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje (ASP o PHP) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL.).

El hecho de que sea estándar no quiere decir que sea idéntico para cada base de datos. En efecto, determinadas bases de datos implementan funciones específicas que no tienen necesariamente que funcionar en otras.

Aparte de esta universalidad, el SQL posee otras dos características muy apreciadas. Por una parte, presenta una potencia y versatilidad notables que contrasta, por otra, con su accesibilidad de aprendizaje.

El SQL trabaja con estructura Cliente / Servidor sobre una red de ordenadores. El ordenador cliente es el que inicia la consulta; el ordenador servidor es que atiende esa consulta. El cliente utiliza toda su capacidad de proceso para trabajar; se limita a solicitar datos al ordenador servidor, sin depender para nada más del exterior. Estas peticiones y las respuestas son transferencias de textos que cada ordenador cliente se encarga de sacar por pantalla, presentar en informes tabulados, imprimir, guardar, etc., dejando el servidor libre.

El SQL permite:

- Definir una base de datos mediante tablas.
- Almacenar información en tablas.
- Seleccionar la información que sea necesaria de la base de datos.
- Realizar cambios en la información y estructura de los datos.
- Combinar y calcular datos para conseguir la información necesaria.

Sus ventajas son:

- Es multiplataforma.
- Es multiusuario.
- Es software libre por lo tanto su costo es nulo.
- Permite manejar la seguridad (decide que usuario se puede conectar al SQL y cuales no).
- Rapidez en el procesamiento de consultas.
- La mayor parte de los gestores de bases de datos se basan en él.

Sus desventajas:

- No se cuenta con soporte, sólo el que se va adquiriendo con la práctica.

2.2.2 ACCESS

Microsoft Access es un Sistema de Gestión de Bases de Datos (DBMS) para uso personal o de pequeñas organizaciones. Es un componente de la suite Microsoft Office aunque no se incluye en el paquete básico. Para bases de datos de gran calibre (en cuanto a volumen de datos o de usuarios) es recomendable usar otros sistemas como Microsoft SQL Server, MySQL u Oracle. Su principal función es ser una potente base de datos, capaz de trabajar en si misma o bien con conexión hacia otros lenguajes de programación, tales como Visual Basic 6.0 o Visual Basic .NET. Pueden realizarse consultas directas a las tablas contenidas mediante instrucciones SQL. Internamente trae consigo el lenguaje Visual Basic for Application (VBA) el cual es similar en forma a VB6.

Permite el ingreso de datos de tipos: Numéricos, Texto, Fecha, Sí / No, OLE, Moneda, Memo y Boolean. Pueden desarrollarse aplicaciones completas basadas en Microsoft Access, pues trae consigo las herramientas necesarias para el diseño y desarrollo de formularios para el ingreso y trabajo con datos e informes para visualizar e imprimir la información requerida.

Su funcionamiento se basa en un motor llamado Microsoft Jet, y permite el desarrollo de pequeñas aplicaciones autónomas formadas por formularios Windows y código VBA (Visual Basic para Aplicaciones). Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Entre las principales funcionalidades de Access se encuentran:

- Crear tablas de datos indexadas.
- Modificar tablas de datos.
- Relaciones entre tablas (creación de bases de datos relacionales).
- Creación de consultas y vistas.
- Consultas referencias cruzadas.
- Consultas de acción (INSERT, DELETE, UPDATE).
- Formularios.
- Informes.
- Llamadas a la API de Windows.
- Interacción con otras aplicaciones que usen VBA (resto de aplicaciones de Microsoft Office, Autocad, etc.).
- Macros.

Además, permite crear programas que muestra la interfaz de usuario, de bases de datos más potentes ya que es un sistema capaz de acceder a tablas externas a través de ODBC[1] como si fueran tablas Access.
Es un software de gran difusión entre pequeñas empresas, cuyas bases de datos no requieren de excesiva potencia, ya que se integra perfectamente con el resto de aplicaciones de Microsoft y permite crear pequeñas aplicaciones con pocos conocimientos de programación.
Entre sus mayores inconvenientes figuran que no es multiplataforma, pues sólo está disponible para sistemas operativos de Microsoft, y que no permite transacciones. Su uso es inadecuado para grandes proyectos de software que requieren tiempos de respuesta críticos o muchos accesos simultáneos a la base de datos.

Sus principales ventajas son:
- Viene incluido en el paquete de Microsoft Office.
- Accesibilidad a un archivo desde el sistema operativo.
- Permite relaciones entre tablas.
- Permite acceder hasta 255 usuarios a la vez.

Sus desventajas son:
- Lentitud en el procesamiento de consultas.
- Sólo funciona en plataforma Windows.
- No admite triggers[2].
- No es seguro.

2.2.3 POSTGRESQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado *Ingres* en la Universidad Berkeley de California. *Ingres* fue más tarde desarrollado comercialmente por la *Relational Technologies/Ingres Corporation*.

[1] **Open DataBase Connectivity**. Es un estándar de acceso a bases de datos que utilizan los sistemas Microsoft. A través de ODBC, en un sistema Windows se puede conectar con cualquier base de datos. sin importar qué Sistema Gestor de Bases de Datos (*DBMS* por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el *DBMS*, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el *DBMS* entienda. Para que esto funcione tanto la aplicación como el *DBMS* deben ser compatibles con *ODBC*, esto es que la aplicación debe ser capaz de producir comandos *ODBC* y el *DBMS* debe ser capaz de responder a ellos.
[2] **Triggers**. objetos relacionados con tablas y almacenados en la base de datos que se ejecutan o se muestran cuando sucede algún evento sobre sus tablas asociadas.

En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres.

En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve periplo como *Postgres95*. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x.

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Altamente Extensible

PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

API Flexible

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java / JDBC, Ruby, TCL, C / C++, y Pike.

Lenguajes Procedurales

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL / pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

MVCC

MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Si alguna vez ha usado algún DBMS con capacidades SQL, tal como MySQL o Access, probablemente habrá notado que hay ocasiones en las una lectura tiene que esperar para acceder a información de la base de datos. La espera está provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL L mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Cliente / Servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Write Ahead Logging (WAL)

La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

Sus principales ventajas son:

- Presenta una mejor escalabilidad y rendimiento bajo grandes cargas de trabajo.
- Ofrece una garantía de integridad en los datos.
- Multiplataforma.
- Existen varias herramientas gráficas de alta calidad para administrar las bases de datos y para hacer diseño de bases de datos.

Sus desventajas son:

- Es un manejador lento para la respuesta de consultas.
- No tiene buena documentación.

2.2.4 ORACLE

Oracle es básicamente una herramienta cliente / servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc.
Oracle como antes se mencionó, se basa en la tecnología cliente / servidor, pues bien, para su utilización primero sería necesario la instalación de la herramienta servidor (Oracle 8i) y posteriormente podríamos atacar a la base de datos desde otros equipos con herramientas de desarrollo como Oracle Designer y Oracle Developer, que son las herramientas básicas de programación sobre Oracle.
Oracle posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Estos porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.

Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.

Oracle provee salvar con seguridad de error lo visto en el monitor y la información de acceso y uso.

Oracle soporta un verdadero ambiente cliente servidor. Este establece un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.

Las principales ventajas de este manejador son:

- Es multiplataforma.
- Al efectuar una operación dentro de la base de datos, esta no es modificada a menos que la operación sea completada. A esto se le llama soporte de transacción.
- Su nivel de fallas es bajo.
- Puede cambiar su tamaño o configuración para adaptarse a circunstancias cambiantes.

Sus mayores desventajas:

- Su mayor defecto es su enorme precio, que varia según versiones y licencias.
- La poca seguridad de su plataforma.

2.3. Teoría básica de programación.

Los lenguajes de programación son la herramienta que nos permite crear software y programas para algún fin determinado. Para entender mejor este concepto se dará un repaso a los antecedentes históricos.

Al desarrollarse las primeras computadoras electrónicas, se vio la necesidad de programarlas, es decir, de almacenar en memoria la información sobre la tarea que iban a ejecutar. En un principio se programaba con lenguaje máquina pero este era muy complicado ya que a cada acción que era capaz de realizar la máquina se le asignaba un número además de estar propenso a errores. A éste le siguió el lenguaje ensamblador el cual es una colección de símbolos mnemónicos que representan: operaciones, nombres simbólicos, operadores y símbolos especiales. Además éste permitía generar el código máquina de manera automática pero requiere de conocer el funcionamiento interno del microprocesador.

Así, nació el concepto de Lenguaje de Alto Nivel, con el primer compilador de FORTRAN (FORmula TRANslation), que, como su nombre indica, inició como un "simple" esfuerzo de traducir un lenguaje de fórmulas, al lenguaje ensamblador y por consiguiente al lenguaje de máquina. A partir de FORTRAN, se han desarrollado innumerables lenguajes, que siguen el mismo concepto: buscar la mayor abstracción posible, y facilitar la vida al programador, aumentando la productividad, encargándose los compiladores o intérpretes de traducir el lenguaje de alto nivel, al lenguaje de máquina.

Existen dos tipos de programación: la lineal y la estructurada.

2.3.1 Programación Lineal

Es aquella que usa un modelo matemático para la descripción de un problema en específico. Trata la planeación para obtener el mejor resultado de las actividades.
En la programación lineal las sentencias se ejecutan secuencialmente mientras están disponibles continuamente todos los recursos de la máquina. Esta secuencialidad se puede alterar con el empleo de instrucciones de modificación de ciclo que permiten dividir un programa lineal en partes o bloques de ejecución condicionada o dependiente de alguna señal, de forma que si ésta no se cumple, el bloque no es ejecutado, con distintos efectos sobre las variables según el tipo de condición empleada.

La programación lineal utiliza un algoritmo, llamado simplex, para obtener una solución sencilla a los problemas planteados.

Este modelo de análisis, naturalmente, se utiliza para resolver muchas otros problemas que admiten una formulación semejante. Se emplea para mejorar la asignación de recursos en empresas que desarrollan un conjunto variado de actividades, para encarar problemas de logística y, en ocasiones, como un modo de hacer más racional los procesos de planificación.

La desventaja ocurre cuando los sistemas se vuelven complejos. Las líneas de código son demasiadas y los programas que eran fáciles se vuelven complicados. A raíz de esta problemática, surgen los lenguajes basados en la programación estructurada.

2.3.2 Programación Estructurada

Se basa en separar de un programa las partes complejas o complicadas en módulos que se comunican entre sí, mandándolos a llamar independientemente para ir resolviéndolos de manera que se vayan requiriendo.

Los módulos al ser independientes, facilitan que si se tienen que borrar o modificar no afecten a las otras estructuras, permitiendo incluso la reutilización de algunos para otros programas.

Con la programación estructurada, elaborar programas de computador sigue siendo una labor que demanda esfuerzo, creatividad, habilidad y cuidado. Sin embargo, con este estilo podemos obtener las siguientes ventajas:

- Los programas son más fáciles de entender, ya que pueden ser leído de forma secuencial, sin necesidad de hacer seguimiento a saltos de línea dentro de los bloques de código para entender la lógica.
- La estructura del programa es clara puesto que las instrucciones están más ligadas o relacionadas entre sí.

- Reducción del esfuerzo en las pruebas. El seguimiento de los fallos o errores del programa se facilita debido a la estructura más visible, por lo que los errores se pueden detectar y corregir más fácilmente.
- Reducción de los costos de mantenimiento de los programas.
- Programas más sencillos y más rápidos (ya que es más fácil su optimización).
- Los bloques de código son auto explicativos, lo que facilita a la documentación.

El principal inconveniente de este método de programación, es que se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático su manejo.

Este tipo de solución, nos lleva a la programación orientada a objetos.

2.3.3 Programación Orientada a Objetos

Simplifica la programación estructurada, dividiendo en pequeñas unidades lógicas de código, que se les llama objetos. Estos tienen la característica de comunicarse entre ellos por mensajes.

Los objetos contienen datos y programas con su propia estructura, formando parte de una organización. Posee atributos (estado) y realiza acciones (comportamiento). Son lo más parecido a objetos reales para convertirlos en objetos de software. Sus atributos los mantiene en variables y lleva a cabo las acciones mediante funciones o métodos.

Las características siguientes son las más importantes:

Abstracción: Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar *cómo* se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.

Encapsulamiento: Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

Principio de ocultación: Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que específica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.

Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o

asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

Herencia: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que reimplementar su comportamiento. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple; esta característica no está soportada por algunos lenguajes (como Java).

2.4. Características, ventajas y desventajas de lenguajes de programación.

2.4.1 JAVA

Este lenguaje fue diseñado por Sun Microsystems Inc. El propósito de la creación de este lenguaje fue que pudiera ser compatible con redes de computadoras de toda clase y que fuera totalmente independiente de la plataforma en que se vaya a manejar.

Sus características son:

Simple: Se encarga de que los lenguajes complicados se queden atrás y su utilización de manera simple es más amigable.

Familiar: Su sintaxis todavía tiene mucho que ver con lenguajes como C o C++, lo que permite al usuario programador familiarizarse en corto plazo.

Robusto: Java maneja la memoria de la computadora, sin que el usuario se tenga que preocupar por su uso.

Seguro: Java cuenta con políticas para que no se pueda infiltrar algún tipo de virus bajo código de java.

Portable: Mientras se cuenta con el intérprete de Java, cualquier código compilado puede ser utilizado en cualquier máquina.

Independiente a la Arquitectura: Java no depende de una arquitectura definida ya que el código es interpretado por cualquier computadora que tenga el intérprete de java.

Multithreaded: Es un lenguaje que puede ejecutar distintas líneas de código al mismo tiempo.

Dinámico: Si un programa cuenta con más de una clase y realizas una modificación en alguna de ellas, Java no requiere compilar todas; basta con que compile la clase modificada y se encarga de hacer una revisión de cada una.

Ventajas	Desventajas
Multiplataforma	Para ser ejecutado requiere de un intérprete.
El JDK(Java Development Kit) de Java es libre. Por lo que hay varios proveedores.	Para el soporte técnico hay varias opciones y es complicado escoger la opción más adecuada.
Sus programas se ejecutan en el servidor.	Corre más lento que si se corriera en el hardware real.
Se puede acceder a bases de datos a través del JDBC (Java Database Connectivity).	Puede ser que no exista un JDBC para bases poco comerciales.
Funciona como una aplicación sola o como un pequeño programa (applet).	
Su vasta colección de bibliotecas estándar.	

2.4.2 VISUAL BASIC

Visual Basic es un lenguaje de programación desarrollado por Microsoft y proviene del lenguaje de programación BASIC (Beginner´s All-purpose Symbolic Instruction Code). Esta diseñado para simplificar la programación utilizando un ambiente grafico para el desarrollo de aplicaciones para Windows.

Características de Visual Basic:

Diseñador de entorno de datos: Automáticamente crea interacción entre controles y datos mediante la acción de arrastrar y colocar sobre formularios o informes.

Los Objetos Activex: Se refiere a la facilidad de acceso a datos a través de la acción de arrastrar y colocar sobre formularios o informes.

Asistente para formularios: Genera de manera automática formularios que permiten administrar registros de tablas o consultas pertenecientes a una base de datos, hoja de cálculo u objeto.

Asistente para barras de herramientas: Permite que el usuario diseñe sus propias barras de herramientas y la elección de los botones que desea visualizar en la ejecución.

Aplicaciones HTML[3]: Visual Basic permite combinar su código con el de HTML para ejecutar una aplicación en una página web.

Ventana de Vista de datos: Nos permite acceder a la estructura de una base de datos. Así como al Diseñador de Consultas y diseñador de Base de datos para administrar los registros.

[3] **HTML**, acrónimo inglés de **H**yper**T**ext **M**arkup **L**anguage (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Ventajas	Desventajas
Amigable con el usuario.	La sintaxis es complicada.
Esta respaldado por Microsoft que es una marca conocida.	No es multiplataforma solo se ejecuta en Windows.
Tiene una ligera implementación de programación orientada a objetos.	La ligera implementación de programación orientada a objetos no permite sacar el máximo provecho de este modelo de programación.
	No permite el manejo de los punteros.
Maneja bases de datos por medio de ADO(Active Data Object), OLE[4] DB u ODBC.	Puede que no exista ODBC para bases de datos poco comerciales.

2.4.3 PHP

PHP es un lenguaje de script embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características especificas de si mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas web.

PHP inició como una modificación a Perl, por el programador Danés-Canadiense Rasmus Lerdorf en el año 1994. El 8 de junio del 1995 fue publicado "**P**ersonal **H**ome **P**age Tools" después de que Lerdorf lo combinara con su propio Form Interpreter para crear PHP / FI .

Dos programadores israelíes del Technion, Zeev Suraski y Andi Gutmans, rescribieron el analizador sintáctico (*parser* en inglés) en el año 1997 y crearon la base del PHP 3.0, cambiando el nombre del lenguaje a la forma actual.

Para 1999, Suraski y Gutmans rescribieron el código de PHP, produciendo lo que hoy se conoce como Zend Engine o motor Zend. En mayo de 2000 PHP 4 fue lanzado bajo el poder del motor Zend Engine 1.0. El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine II (o Zend Engine 2). La versión más reciente de PHP es la 5.1, que incluye el novedoso PDO (Objetos de Datos de PHP o PHP Data Objects) y mejoras utilizando todas las ventajas que provee el nuevo Zend Engine 2. En la actualidad se cuenta con PHP 5.1.6 pero no ha sufrido grandes mejoras.

Gracias a que PHP utiliza scripts CGI[5], podemos realizar el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas; donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa. Un sitio con páginas dinámicas es aquel que nos permite tener comunicación con el visitante.

[4] **Object Linking and Embedding** (Incustracón y Vinculación de Objetos) Es la tecnologia que permite crear aplicaciones que contengan llamadas, codigo o cualquier componente de otra aplicación.

[5] **Scripts CGI (Common Gateway Interface)**, el CGI es un programa de interfaz que permite al servidor de Internet utilizar programas externos para realizar una función específica, estos consisten generalmente de una serie de instrucciones escritas en un lenguaje de programación como C o PERL que procesan la petición de un navegador, ejecutan un programa y formatean los resultados en HTML de manera que puedan ser presentados en el navegador.

Sus características son:

Soporte para gran cantidad de manejadores de bases de datos: Entre su soporte pueden mencionarse InterBase, SQL, MySQL, Oracle, Informix, PostgreSQL, entre otras.

Bibliotecas externas: Ayudan al programador ha generar desde documentos en pdf hasta analizar código XML.

Sintaxis similar a la de ASP(Active Server Pages): Pues el código PHP va incrustado dentro del código HTML.

Manejo de páginas dinámicas: Ofrece una solución simple para la paginación dinámica en la Web.

Código abierto: Nos permite realizar mejoras en el funcionamiento y reparar errores en el código.

Multiplataforma: PHP nos brinda la opción de ejecutarlo en cualquier plataforma de UNIX o de Windows, con el software de Netscape o del Web Server de Microsoft.

Soporte de sintaxis orientado a objetos y una sintaxis de lenguaje mucho más potente y consistente.

Extensibilidad: Provee a los usuarios finales una infraestructura para muchísimas bases de datos, protocolos y APIs.

Ventajas	**Desventajas**
Fácil manejo.	
Soporta en cierta medida la orientación a objeto.	La orientación a objetos es aún muy deficiente para aplicaciones grandes.
El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática.	
Se puede incrustar código PHP con etiquetas HTML.	La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
Excelente soporte de acceso a base de datos.	
La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.	Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

2.5. Elección óptima de solución para nuestro sistema.

Como ya se había mencionado en los requerimientos del sistema decidimos crear un sistema de consulta, donde se tenga acceso a la información requerida por los usuarios.

Dicho sistema se pondrá en un sitio web por lo que necesitamos un lenguaje que sea compatible con HTML y que a la vez facilite el desempeño del manejador de base de datos, para no crear complicaciones al personal encargado del mantenimiento de este sistema.

Después de analizar las características de cada uno de los lenguajes de programación así como de los manejadores de bases de datos se llegó a las siguientes conclusiones.

PHP es el lenguaje que nos da la mejor opción para el desarrollo del sistema ya que se puede obtener en la web y su código esta disponible bajo la licencia GPL[6], al poderse encapsular dentro de código HTML se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente, viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor.

Para poder trabajar con PHP es necesario contar con un servidor virtual en este caso se utilizará Apache[7].

En cuanto a los manejadores de bases de datos, se hicieron varias pruebas, ya que la base de datos se encuentra en Access, fue necesario migrar algunos datos a postgreSQL, para su uso dentro del servidor de la DIE, así como pasar la base a formato .CVS (separación por comas) para que postgreSQL, pudiera leer la base y así trabajar con ella, proporcionando rapidez en el procesamiento de consultas.

Este manejador de base de datos, utiliza lenguaje SQL, por lo que nos lleva a la misma comparación, que ya se menciono, con los otros manejadores.

Así, PHP y postgreSQL proporcionan las características necesarias para el cumplimiento de nuestro objetivo y el desarrollo del sistema.

.

[6] La **GPL** (*General Public License* o licencia pública general) es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

[7] El **servidor HTTP Apache** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD , GNU / Linux, etcétera), Windows y otras, que implementa el protocolo HTTP / 1.1 y la noción de sitio virtual.

Capítulo 3

Diseño e implementación del sistema

Capítulo 3. Diseño e implementación del sistema

Los sistemas de información deben de optimizar los recursos con que se cuente para cubrir las necesidades de los requerimientos de los usuarios. Para ello se emplea la ingeniería de programación, la cual se apoya e incluso desarrolla lenguajes de programación enfocados a facilitar el desarrollo de los sistemas de información.

3.1. Análisis del sistema.

3.1.1 Antecedentes

El análisis de sistemas se trata básicamente de investigar los detalles del sistema de información a desarrollar, es decir, determina los objetivos y límites del sistema, caracteriza su estructura y funcionamiento, marca las normas que permitan alcanzar los objetivos propuestos y evalúa sus consecuencias. Dependiendo de los objetivos del análisis podemos encontrarnos ante dos problemáticas distintas:

- Análisis de un sistema ya existente para comprender, mejorar, ajustar y/o predecir su comportamiento.
- Análisis como paso previo al diseño de un nuevo sistema-producto.

El sistema que se va a desarrollar se enfocará a mantener informado al personal académico de la DIE, para que puedan consultar el estado de su contrato, las fechas de inscripción y realización de exámenes extraordinarios, así como las fechas para entregar documentación en la dirección y en la secretaría auxiliar de la DIE. También se deberá proporcionar la documentación que necesita el personal de nuevo ingreso para formar parte de la DIE.

Para que el personal académico pueda tener acceso al sistema, tendrá que proporcionar su RFC para ser validado con la base de datos e identificar si es correcto o no, en caso de que no estuviera contenido en la base, podrá mandar un correo a través de una liga que lo llevará a otra pantalla.

Cuando el usuario este dentro del sistema, podrá ver las fechas del contrato, así como el nombramiento actual, si el usuario quisiera, podrá ver a partir de una liga, las fechas de inscripción y su realización de exámenes extraordinarios.

Si el usuario desea ver las fechas de Consejo Técnico, podrá verlo a través de otra liga que se desarrollará para que las consulte (fecha de entrega de documentación en la secretaría auxiliar o bien en la dirección).

En caso de ser personal de nuevo ingreso se proporcionará una liga que mostrará la documentación que éste necesita para formar parte de la DIE, esto sin necesidad de ingresar su RFC, ya que por ser de nuevo ingreso no está en la base de datos.

Cuando los requerimientos estén revisados, se empezará a diseñar el sistema con base a todos las necesidades ya mencionados y quedaría de esta manera:

El sistema requiere ingresar a una página que cuente con tres opciones:

Consulta del estatus del personal académico.
Consulta del calendario de periodos de exámenes extraordinarios.

Consulta del calendario de fechas de Consejo Técnico.

Contamos con 2 usuarios:

El personal académico, que son las personas que harán uso del sistema.

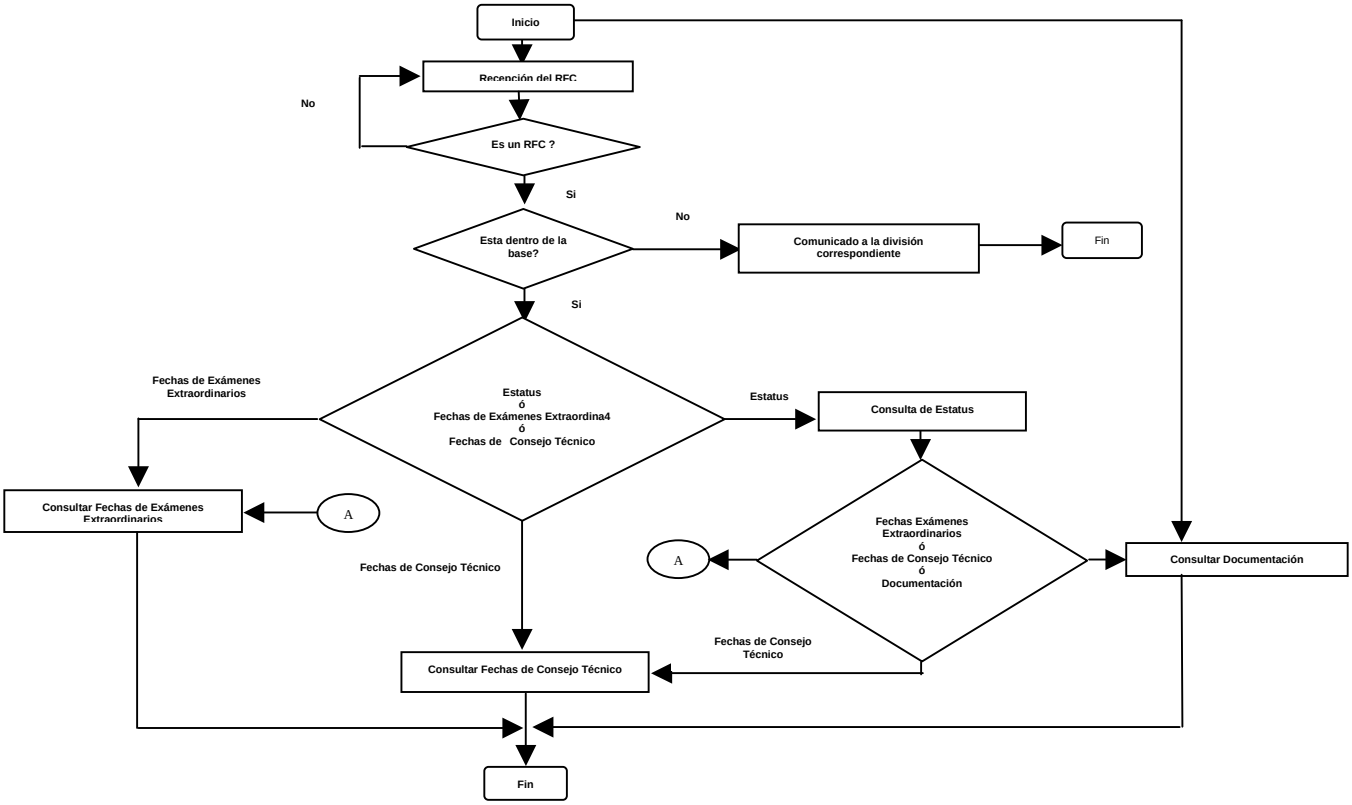


Figura 5. Diagrama de Flujo para el Usuario

El personal académico tendrá acceso al sistema al ingresar su RFC para que este se valide en la base de datos, si es correcto podrá consultar cualquiera de las tres opciones, examinaremos cada una de ellas para este usuario:

1. Si el académico selecciona entrar a consultar su estatus.

1.1 La base da datos arrojará sus datos personales así como su estatus, en caso de que se detecte algún problema, el usuario podrá mandar un mail al administrador para que se corrija el error.
1.2 Además se le proporcionarán las opciones para consultar los periodos de exámenes extraordinarios así como las fechas de Consejo Técnico, sin necesidad de regresar al inicio.
1.3 En caso de requerir hacer algún trámite, éste podrá acceder a una página que cite la documentación y pasos a seguir para dicho trámite.

2. Si se selecciona la opción de calendario de periodos de exámenes extraordinarios, podrá consultarlos directamente.

2.1 Sólo estará disponible el semestre en curso tanto para la opción de inscripciones como para la de realización.

3. Si selecciona calendario de fechas de Consejo Técnico, entrará directamente a la página.

3.1 Podrá consultar las fechas de Consejo Técnico para el semestre en curso.

Por otra parte si el personal es de nuevo ingreso podrá consultar una página donde se citará la documentación necesaria para que este forme parte de la DIE (No será necesario que ingrese su RFC).

Administrador(es), que es la persona que mantendrá el sistema actualizado.

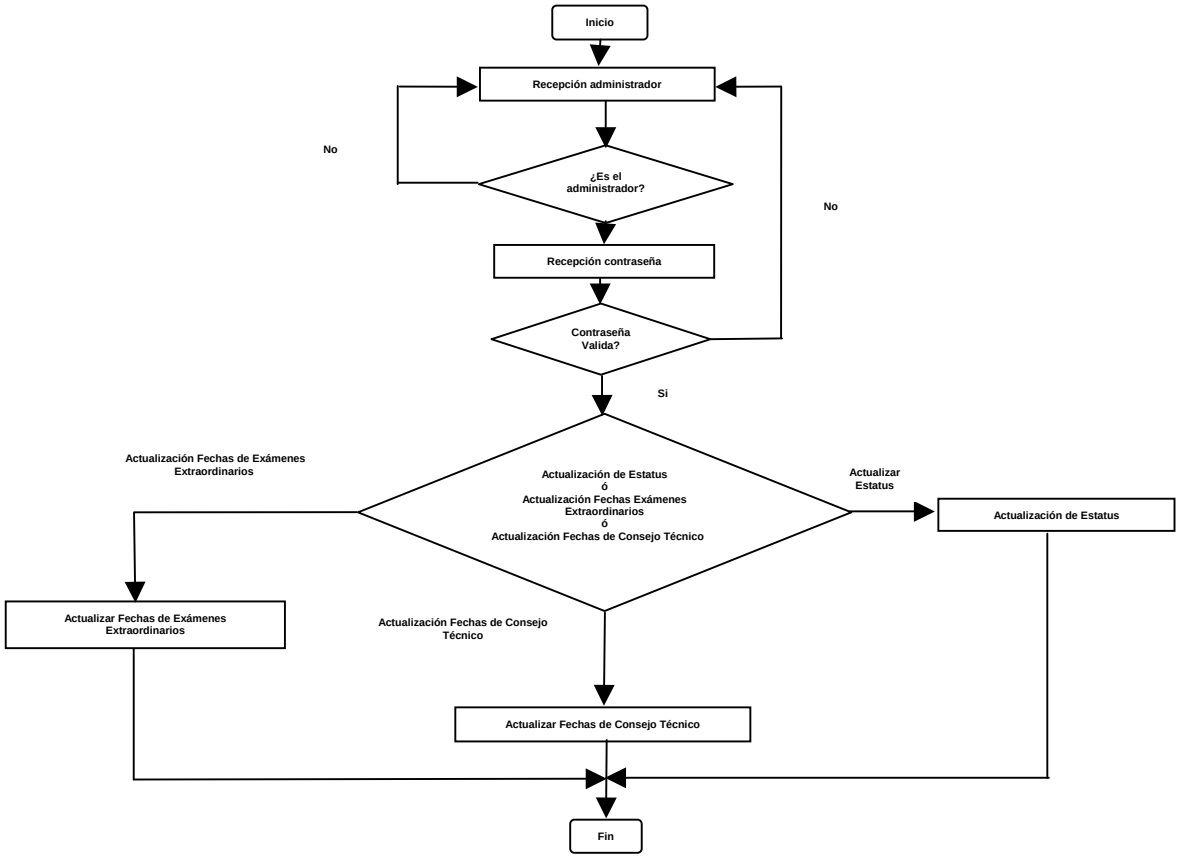


Figura 6. Diagrama de Flujo para el Administrador

El administrador tendrá acceso a las tres funciones principales del sistema, ya que es un empleado de la DIE, así mismo deberá darle el mantenimiento necesario al sistema, para que éste se mantenga actualizado.

1. En cuanto a las funciones del sistema el administrador tendrá que validarse a través de una contraseña.
2. Para mantener actualizada la opción de consulta de estatus el administrador deberá mantener la base de datos actualizada después de cada periodo de Consejo Técnico.
3. Para las opciones de consulta de calendario de Consejo Técnico y consulta de periodos de exámenes el administrador deberá cambiar esta información por semestre.

Debido a que el administrador tendrá que cambiar las fechas de exámenes extraordinario y las de Consejo Técnico, lo más viable será diseñar una pequeña base de datos que contenga dicha información así como la contraseña para que el administrador pueda actualizar el sistema.

3.2. Diseño del sistema.

3.2.1 Antecedentes

El diseño de un sistema es un proceso y un modelado a la vez. El proceso de diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del sistema a construir. A lo largo del diseño se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas:

- El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente.
- Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el sistema.
- El diseño debe proporcionar una completa idea de lo que es el sistema, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación.

El proceso de diseño de un sistema exige buena calidad a través de la aplicación de principios fundamentales de diseño, metodología sistemática y una revisión exhaustiva.

Cuando se va a diseñar un sistema se debe tener presente que el proceso de un diseño incluye, concebir y planear algo en la mente, así como hacer un dibujo o modelo o croquis.

3.2.2 Diagrama del Sistema para el Personal Académico

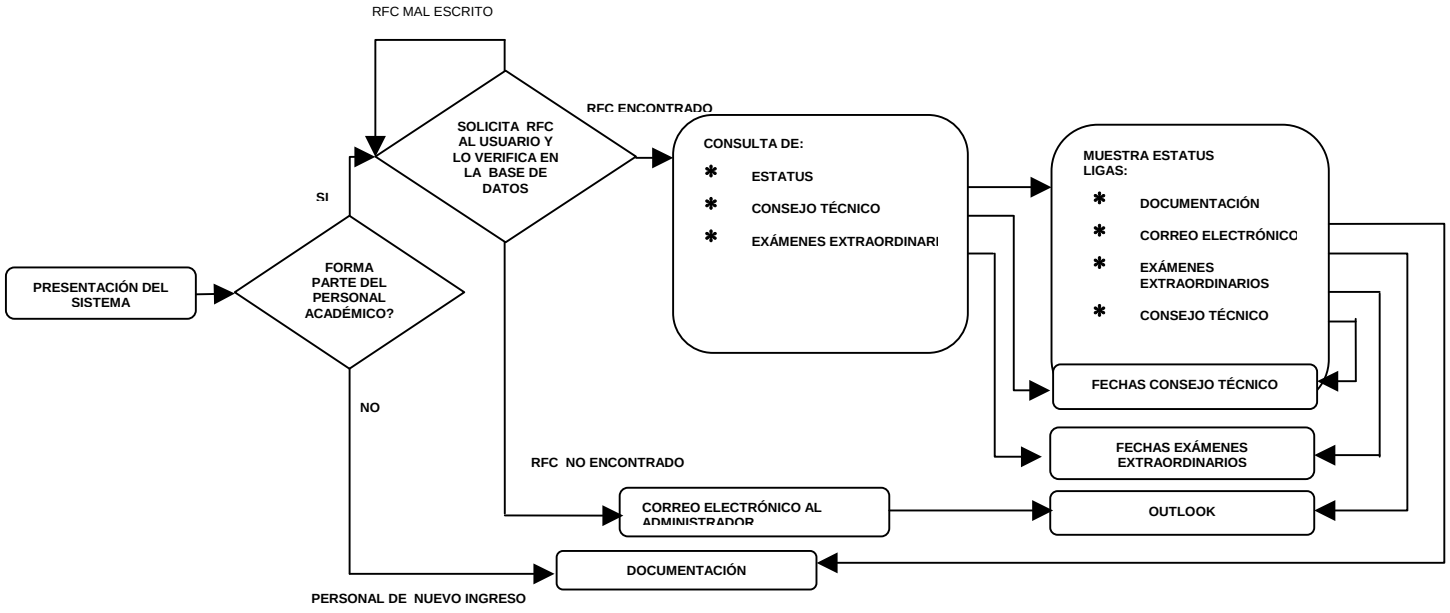


Figura 7. Diagrama del Sistema para el Personal Académico.

3.2.3 Procesos del Sistema

Como ya habíamos mencionado el sistema consta de varios procesos, los cuales se diseñaran por medio de un diagrama.

3.2.3.1 Proceso Validación RFC

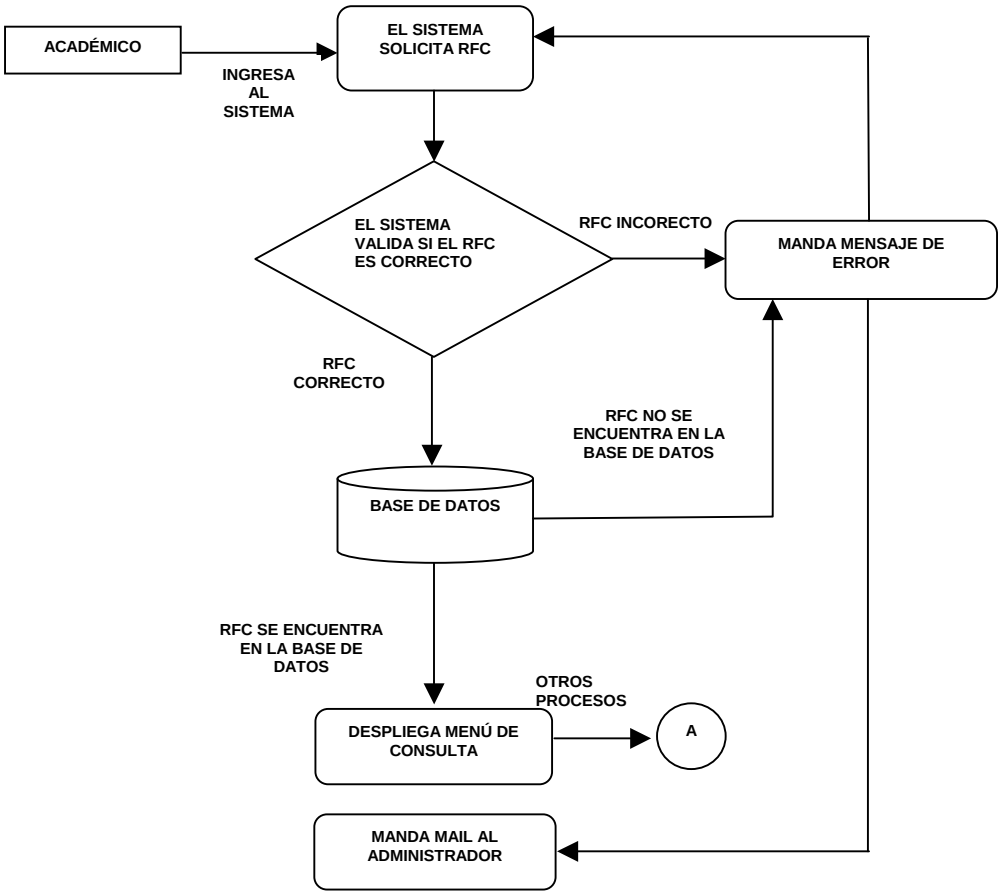


Figura 8. Diagrama Proceso de Validación RFC.

3.2.3.2 Proceso de Consulta de Estatus

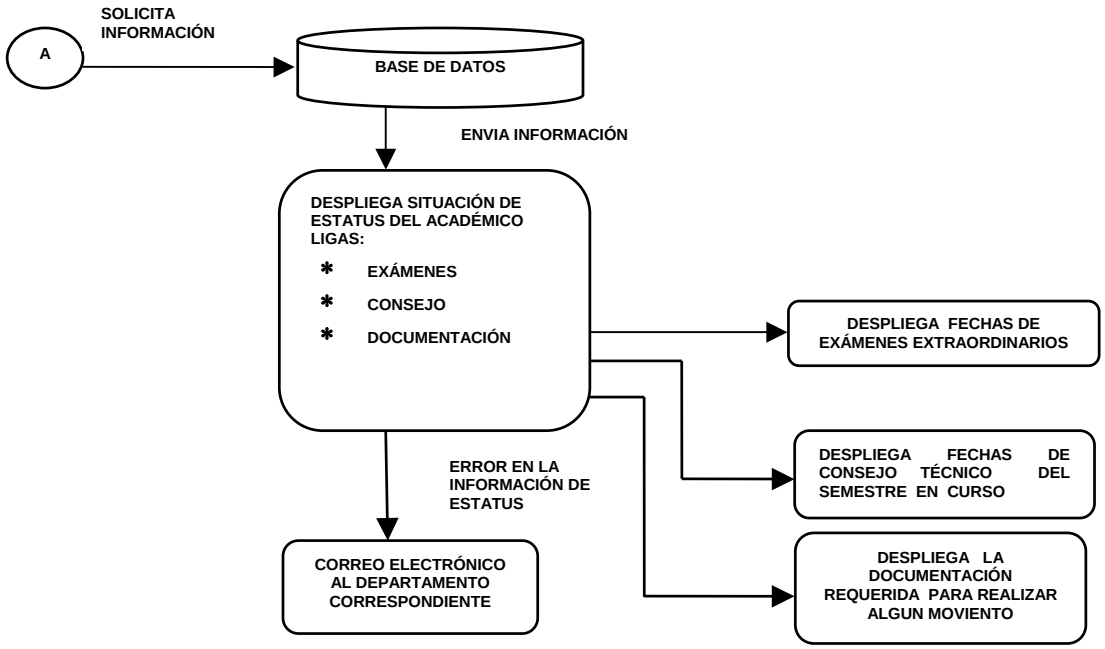


Figura 9. Diagrama Proceso de Consulta de Estatus.

3.2.3.3 Proceso de Fechas de Exámenes Extraordinarios

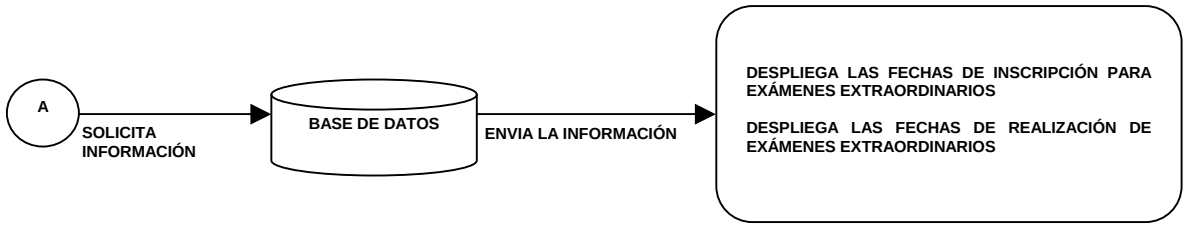


Figura 10. Diagrama de Fechas de Exámenes Extraordinarios.

Si el académico selecciona la opción fechas de exámenes extraordinarios del menú principal, el sistema solicitará dicha información a la base de datos y ésta se desplegará en la pantalla.

3.2.3.4 Proceso de Fechas de Consejo Técnico

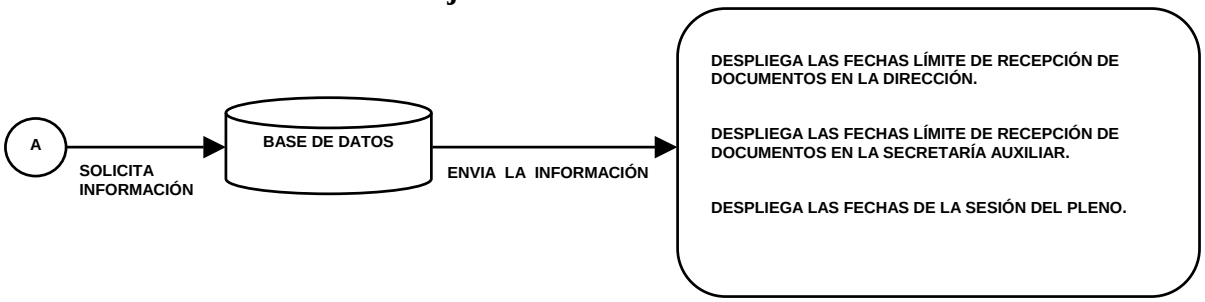


Figura 11. Diagrama de fechas de Consejo Técnico.

Si el académico selecciona la opción fechas de Consejo Técnico del menú principal, el sistema buscará dicha información en la base de datos y se desplegará una pantalla con esta información, correspondiente al semestre en curso.

3.2.3.5 Proceso Documentación

Figura 12. Diagrama Documentación.

Si el académico selecciona la opción de documentación de la página de estatus, inmediatamente se le desplegará una pantalla con dicha información.
De igual manera lo hace al ingresar, si el RFC no se encuentra también podrá consultarla.

3.2.4 Diagrama del sistema para el administrador

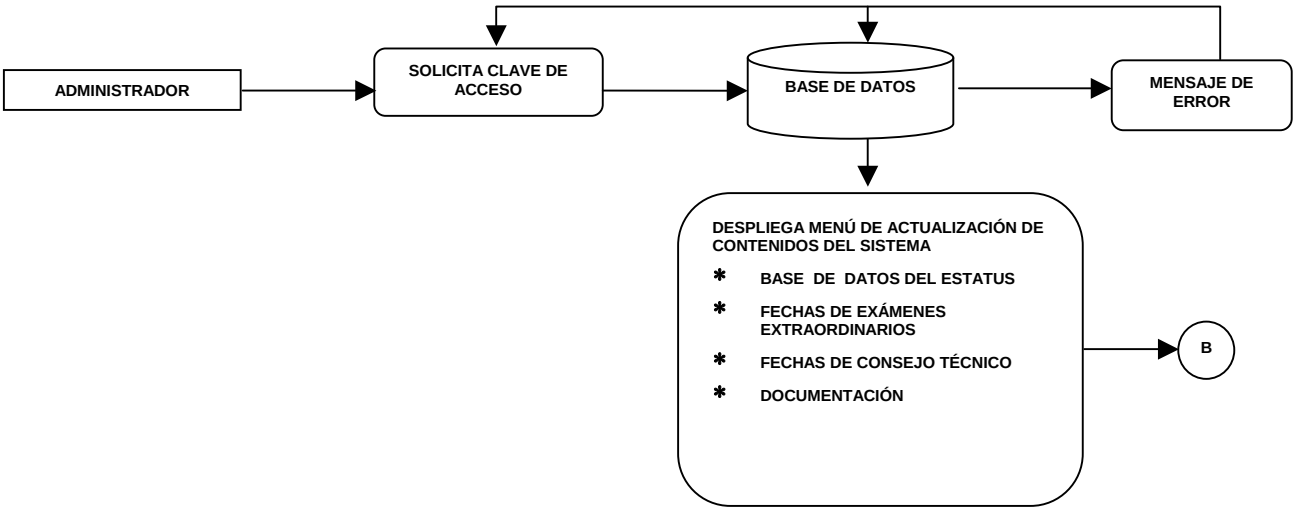


Figura 13. Diagrama del sistema para el administrador.

3.2.4.1 Proceso de Actualización de fechas de Exámenes Extraordinarios

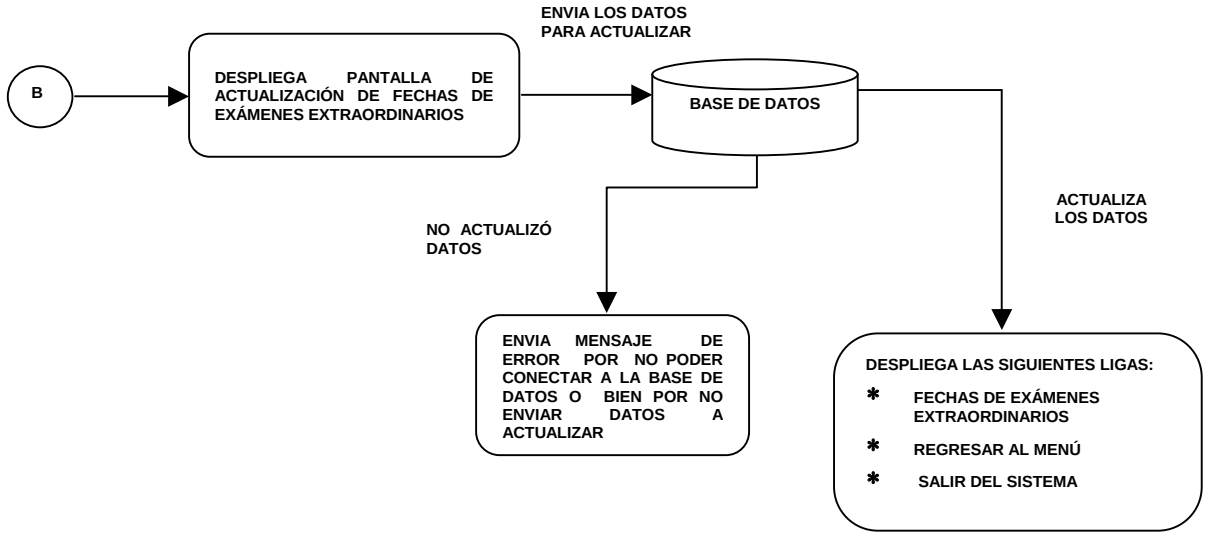


Figura 14. Diagrama Proceso de Actualización de Fechas de Exámenes Extraordinarios.

3.2.4.2 Proceso de Actualización de fechas de Consejo Técnico.

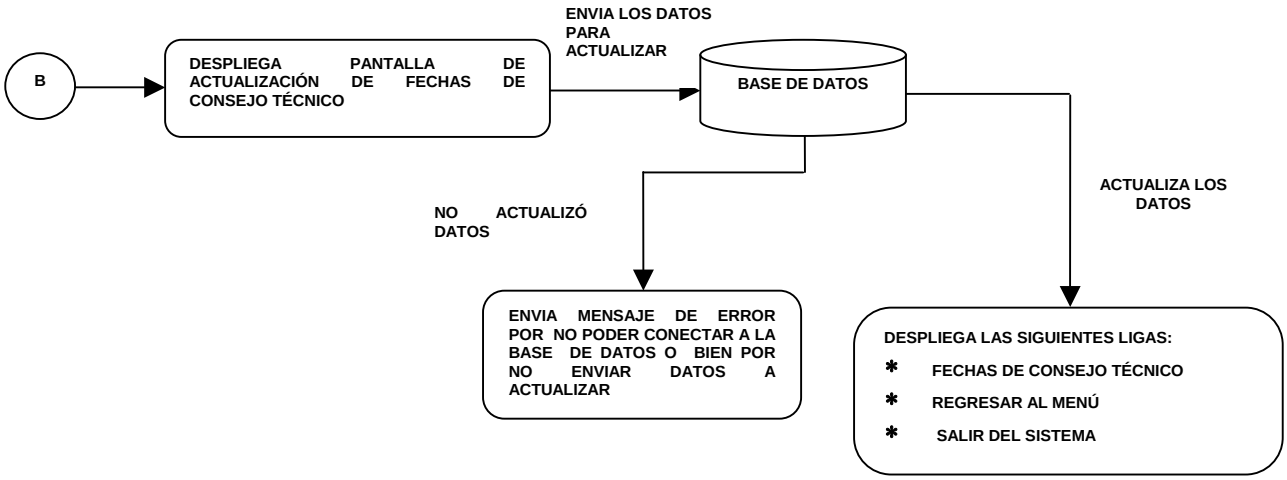


Figura 15. Diagrama Proceso de Actualización de Fechas de Consejo Técnico.

3.2.4.3 Proceso de Actualización Base de Datos Estatus.

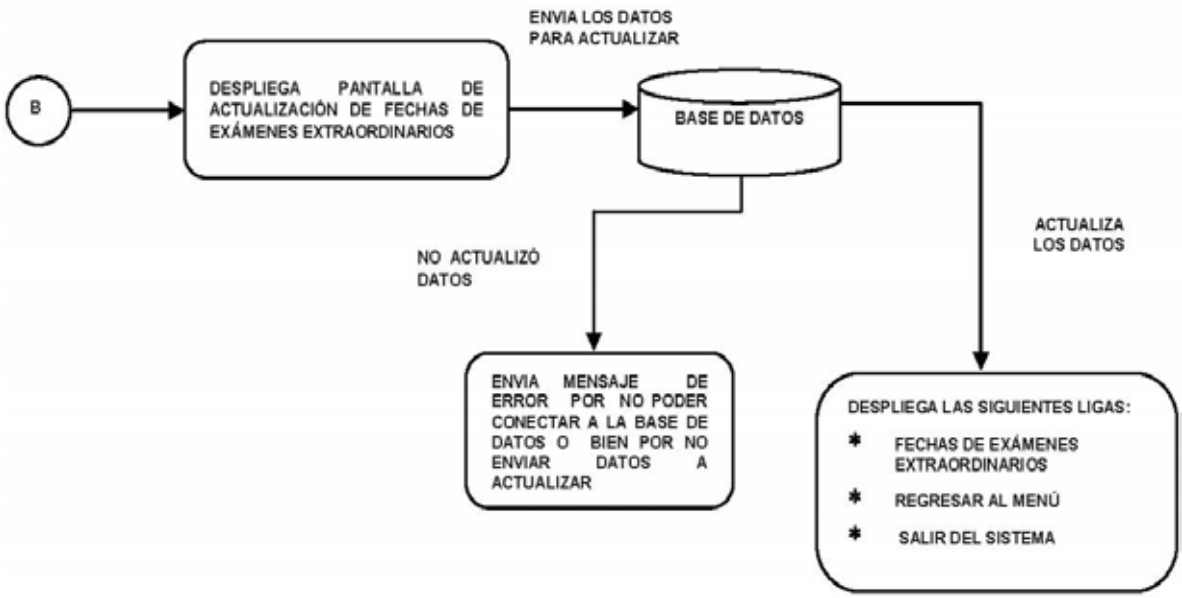


Figura 16. Diagrama Proceso de Actualización Base de Datos Estatus.

3.3. Desarrollo del sistema.

El desarrollo de un sistema tiene como objetivo principal la traducción de las especificaciones de diseño a código fuente. Este objetivo puede alcanzarse haciendo el código fuente tan claro y sencillo como sea posible, de manera que se facilite la depuración, pruebas y modificaciones.

Para este sistema se eligió PHP como lenguaje de programación ya que nos permite embeber pequeños fragmentos de código dentro de la página HTML y realizar de forma fácil y eficaz la presentación del sistema. PHP utiliza las etiquetas <? y ?> para indicar el inicio y el fin de un código respectivamente.

Ejemplo de un código PHP embebido:

```
<html>
<head>
<title>Título</title>
</head>
<body>
Mi primera página en PHP<br><br>
<?
$numero = 10;
while ($numero > 0) {
```

```
echo "El número es: $numero<br>“;
$numero--;
}
?>
</body>
</html>
```

PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones. Como ya se había mencionado el sistema debe trabajar con bases de datos realizadas en Access y PHP por lo tanto, explicaremos la forma de realizar la migración de una tabla con postgreSQL.
Lo primero que se debe hacer es abrir la tabla en Excel, para que al guardar le demos lo opción de .CVS, que normalmente se le conoce como separación por comas, ya que esta extensión nos permite leerla desde el manejador PostgreSQL, que tendrá la conexión al sistema.

El código para conectarse con postgreSQL es:

```
pg_connect("host=Nombre Host dbname=BaseDatos user=Usuario
password=Contraseña")
```

Cuyos parámetros, pasados como una cadena única, indican el nombre del servidor -o IP del mismo- ‘NombreHost’ donde se encuentra la base de datos, el nombre de la base de datos ‘BaseDatos’, el ‘Usuario’ de acceso a la base de datos, y la ‘Contraseña’ de acceso. En caso de éxito la función devuelve un identificador del enlace con el sistema de bases de datos.

Para efectuar consultas sobre una base de datos PostgreSQL, se utiliza en PHP la función:

```
pg_query($conexion, $sql)
```

Que toma como parámetros, el enlace con la base de datos y una cadena con la consulta SQL a ejecutar (SELECT, INSERT, DELETE, etc.). Devuelve un identificador del resultado en caso de éxito o FALSE en caso de error en la consulta.

Solo con la ejecución de la consulta sobre la base de datos, no podemos presentar el resultado de la misma. Para poder mostrar información resultante de una consulta deberemos hacer uso de funciones complementarias. Una de las posibles es:

```
pg_fetch_array($id_resultado, $fila)
```

Que devuelve la fila ‘$fila’ en forma de un arreglo con el resultado de la sentencia extraída identificada por el parámetro ‘$id_resultado’.

Ejemplo:

```
<?php

#Conectamos con PostgreSQL

$conexion = pg_connect("host=192.168.0.3 dbname=BaseDatos user=Usuario
```

```
password=Contraseña") or

die ("Fallo en el establecimiento de la conexión");

#Efectuamos la consulta SQL

$result = pg_query ($conexion, "select * from personal" )

or die("Error en la consulta SQL");

#Mostramos los resultados obtenidos

$i=0;

while( $row = pg_fetch_array ( $result,$i )) {

   echo $row [ "id" ];

   echo $row [ "nombre" ];

   $i++;

}

?>
```

De esta manera, el usuario ve la pantalla mostrándole sus datos, a través de postgreSQL.

Capítulo 4

Pruebas

Capítulo 4. Pruebas

4.1 Introducción

El objetivo de la fase de pruebas de un programa es el de detectar todo posible malfuncionamiento antes de que entre en producción. Un error detectado en el laboratorio puede ser costoso de reparar; pero siempre es peor que el error le aparezca al usuario final.

En esta idea, una batería de pruebas será de mayor calidad cuantos menos errores queden por descubrir tras haberla pasado. Y, viceversa, si un programa aún tiene muchos fallos tras haber superado una batería de pruebas, diremos que esta batería es de poca calidad.

Si pudiéramos probar un programa con todos los posibles datos de entrada, tendríamos una batería de pruebas perfecta, pues no hay lugar para las sorpresas. Lamentablemente, casi nunca es posible probar con todos los casos. En consecuencia necesitamos un criterio para elegir qué casos probamos.

4.2. Pruebas de caja negra

En estas pruebas es recomendable tener el manual de usuario, con las especificaciones de *lo que el programa tiene que hacer*. Con la finalidad de realizar al menos una prueba de todas y cada una de las cosas que el programa tiene que hacer.

Para ello, si le pidió a varios usuarios, que probarán el funcionamiento del sistema.

El primer paso, fue que escribieran su RFC dentro del formulario, de esta manera se comprobó que el sistema hizo las validaciones correspondientes, que se explicaran a continuación:

- Verifica que se teclee el formato del RFC (cuatro letras y seis números).
- Valida el RFC sin importar si lo escriben con minúsculas o mayúsculas.
- En caso de error le da la opción de regresar y volver a ingresar su RFC.

El segundo paso, fue que el sistema reconociera al usuario dentro de la base de datos para lo cual se dieron las siguientes situaciones:

- Cuando el usuario no estuvo dentro de la base de datos, se le sugirió mandar un correo informándole a su división de la situación, desde aquí a través de una liga el usuario pudo ver la documentación de algún trámite.
- Al identificar al usuario dentro de la base le mostró la pantalla que despliega el menú.

De las opciones del menú, la primera es el estatus que al ser elegida, le presentó la situación de su contrato dentro de la DIE, en esta pantalla le permitió ir a la liga de documentación, con la cual, obtuvo la información necesaria, para saber que formato o documentos, requiere para realizar algún trámite.

Al elegir la opción de fechas de exámenes extraordinarios, le presentó una tabla que muestra las fechas de exámenes extraordinarios correspondientes al semestre.

De igual manera al elegir la opción de fechas de Consejo Técnico, le presentó una tabla que muestra las fechas de Consejo Técnico correspondientes al semestre.

Al término de estas pruebas se concluye que el sistema funciona correctamente.

4.3. Pruebas de caja blanca

Si ya se han realizado las pruebas de caja negra de manera satisfactoria, será necesario hacer una revisión del listado del programa para determinar qué porcentaje de líneas de código han sido ejecutadas alguna vez, porcentaje que se conoce como cobertura de sentencias.

En la primera parte, se pide que el dato sea un RFC por medio del formulario, donde empieza la validación del formato del mismo, si no es ese dato, manda un error y permite regresar a la pagina del formulario, compara lo que se escribe con lo que se tiene en la base de datos y lo identifica, si el usuario no estuviera dentro de la base, le permite ir a una liga para avisar por medio de un correo, la situación en el Departamento que le corresponda, a través de una liga le permite ver en otra pantalla la documentación por medio de un enlace para asegurarse de los documentos que tiene que presentar conforme al trámite, personalmente en ventanilla.

Si el usuario pertenece a la base de datos, lo lleva automáticamente a la página del menú, tomando de la base el nombre y le despliega tres opciones, las cuales son:

- Estatus del contrato
- Exámenes Extraordinarios
- Fechas de Consejo Técnico

Cuando el usuario decide ver la opción del estatus del contrato, gracias a la base de datos, toma la información más importante de la misma, a través de PostgreSQL, desplegándola en una tabla y mostrándosela al usuario.

Si el usuario elige Exámenes extraordinarios, por medio de una base de consulta, se descargan las fechas acorde al semestre en curso. La misma situación se da, cuando el usuario selecciona Fechas de Consejo Técnico.

Si el usuario es el administrador le pide la contraseña para ingresar al menú, en caso de que no fuera la contraseña correcta le manda un mensaje de error, sin permitirle acceder.

Si la contraseña es correcta le permite ingresar al menú, que tiene las mismas opciones que ve el usuario, por medio del cual, puede elegir si va a modificar o actualizar algún dato.

Conclusiones

Conclusiones

En la actualidad la DIE, como ya se había mencionado, no contaba con algún medio electrónico que facilitará la comunicación entre su personal y la división correspondiente, lo que ocasionaba serios problemas en tiempo, esfuerzo y recursos.

El objetivo de nuestra tema fue que el personal académico tuviera un sistema que fuera fácil y amigable, para poder obtener información acerca del estatus de su contrato dentro de la Facultad de Ingeniería, así como los periodos tanto de exámenes extraordinarios y los de Consejo Técnico para poder llevar a cabo cualquier trámite, sin tener que recurrir a la Secretaría, es por eso que se creo el Sistema de “Consulta de Información de Estatus del Personal Académico y Periodos de Examen por Semestre de la DIE”.

Dicho sistema cumple con la finalidad de proporcionarle al personal académico, sus datos, referentes a su situación en la Facultad de Ingeniería a través de una consulta a la base de datos de la DIE, a la que se tuvo acceso por medio de conexión a PostgreSQL, se estudiaron varios lenguajes de programación y se hizo el estudio detallado para poder elegir cual era el óptimo para el personal académico así como para los administradores del sistema y que no tuvieran problemas en cuanto al manejo y funcionamiento del sistema.

Se llevaron a cabo las pruebas suficientes para comprobar el funcionamiento adecuado del sistema y poder corregir los errores durante este periodo, para que ahora el sistema este disponible.

Por lo tanto, se cumple el objetivo del tema, de proporcionar la herramienta necesaria para poder consultar información importante para el personal académico, por medio de un sistema alojado en el servidor de la DIE, que se puede consultar en cualquier lugar que tenga conexión a Internet.

Anexos

Anexos

Manual de Usuario.

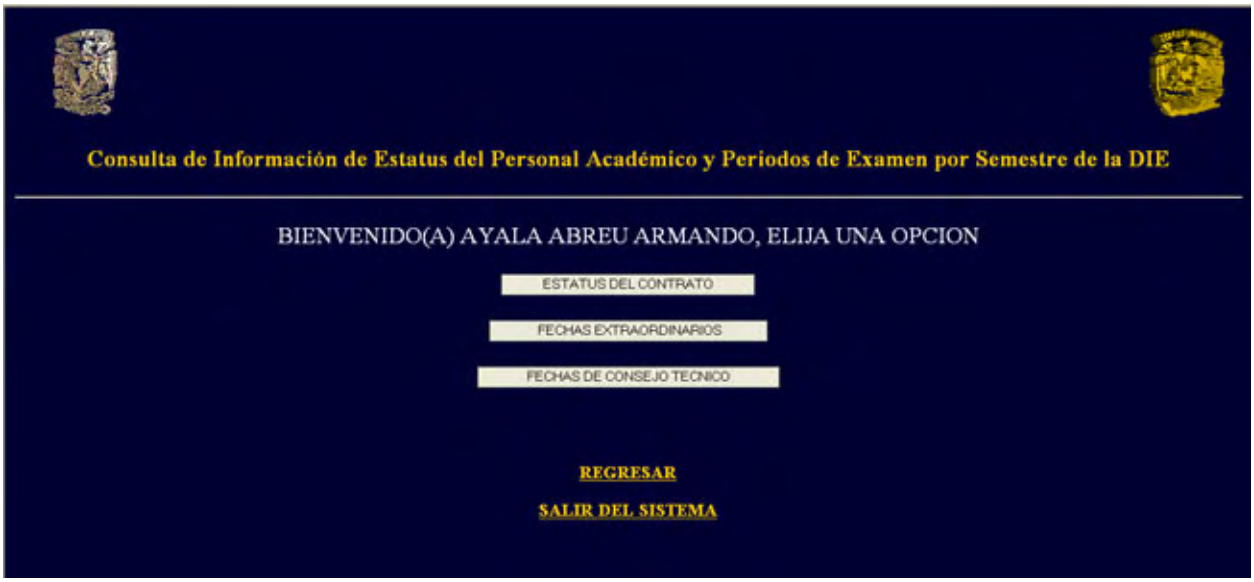
El manejo del sistema “Consulta de Información de Estatus del Personal Académico y Periodos de Examen por Semestre de la DIE” esta diseñado para que el usuario pueda operarlo sin ningún problema.

Para que tenga acceso al sistema de consulta deberá ingresar su RFC, si usted no es parte del personal académico de la DIE, podrá consultar la documentación que le será requerida para formar parte de la DIE, en caso de ser parte del personal deberá colocar su RFC en la casilla correspondiente (RFC consta de cuatro letras y seis números las letras pueden ser minúsculas o mayúsculas).

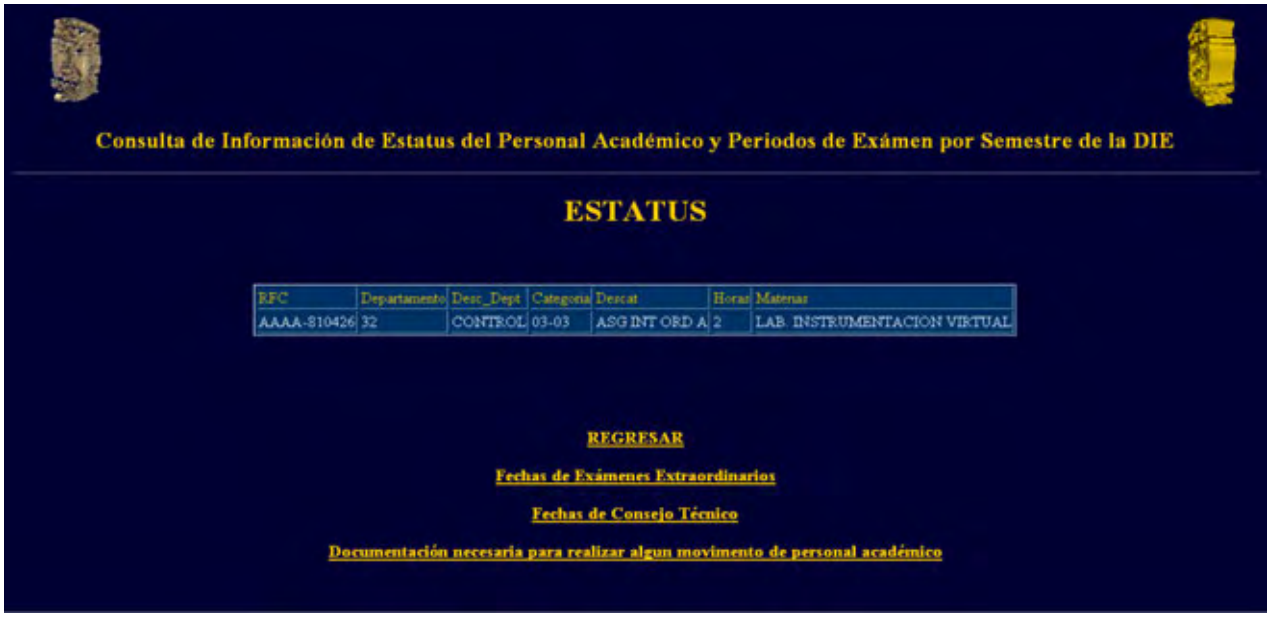
Si no se escribiera correctamente el RFC marcará error al escribir su RFC y tiene la opción de volver a ingresar su RFC.

Si la validación del RFC no fuera exitosa, es decir, que no se encontrará dentro de la base de datos podrá enviar un mail al jefe de departamento correspondiente y este le notificará al administrador del sistema.

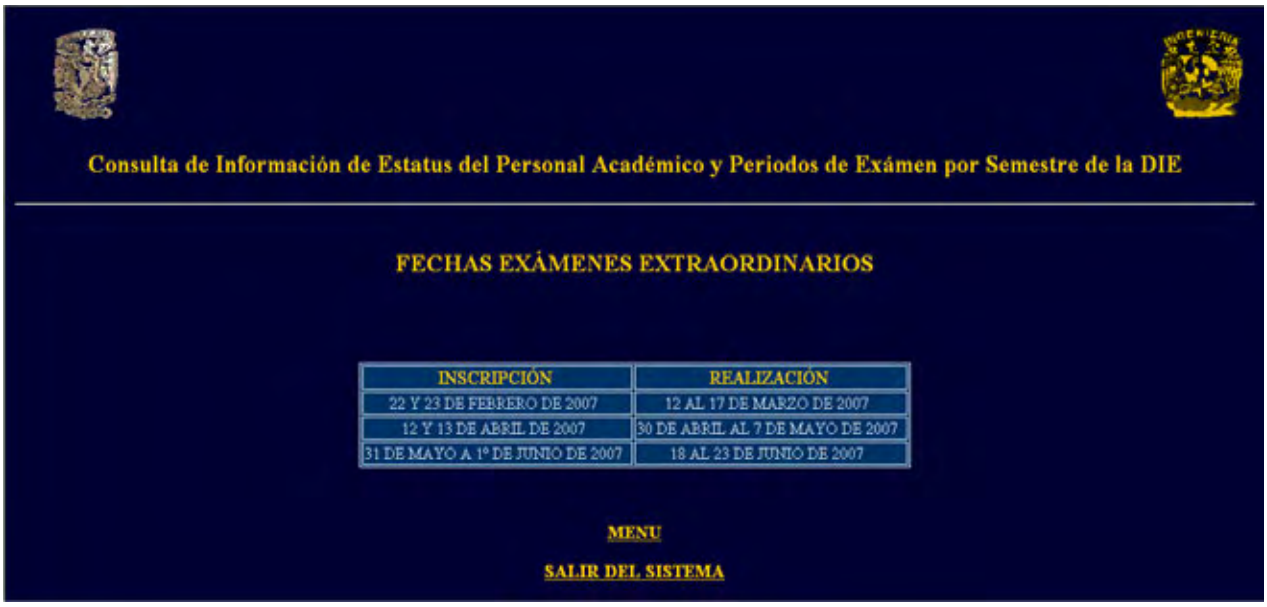
Si la validación del RFC es exitosa el sistema le mostrará la pantalla de MENÚ en la cual verá sus opciones.



Si su elección es Estatus del Contrato le mostrará la pantalla con dicha información.



Esta pantalla también le da la opción de regresar al menú o consultar la documentación para realizar algún trámite, o bien de consultar los periodos de exámenes extraordinarios o los periodos de Consejo Técnico.
Además si algún dato estuviera incorrecto en su Estatus podrá enviar un correo electrónico al departamento correspondiente.
Si eligiera la opción Fechas de Extraordinarios le mostrará una pantalla con los periodos de exámenes extraordinarios.



En esta pantalla se tiene la opción de regresar al menú o salir del sistema.

Si eligiera la opción Fechas de Consejo Técnico le mostrará una pantalla con los periodos de Consejo Técnico.

Consulta de Información de Estatus del Personal Académico y Periodos de Examen por Semestre de la DIE

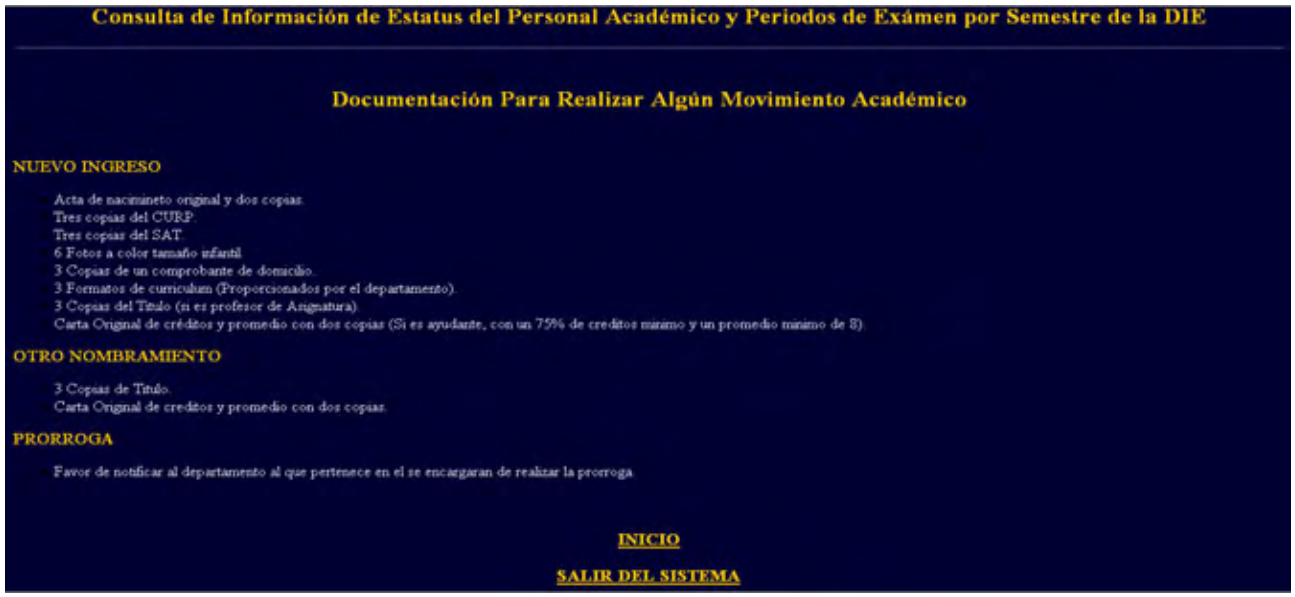
FECHAS DE CONSEJO TECNICO

TIPO	LÍMITE DE RECEPCIÓN DE DOCUMENTOS EN LA DIRECCIÓN	LÍMITE DE RECEPCIÓN DE DOCUMENTOS EN LA SECRETARIA AUXILIAR	SESIÓN DEL PLENO
ORDINARIA	11-Ene-06	15-Dic-05	Jueves 26-Ene-06*
EXTRAORDINARIA			Viernes 24-Feb-06*
ORDINARIA	06-Mar-06	06-Feb-06	Miercoles 29-Mar-06*
EXTRAORDINARIA			Miercoles 03-May-06*
ORDINARIA	03-May-06	03-May-06	Jueves 8-Jun-06 *
EXTRAORDINARIA			Jueves 01-Jun-06 *
ORDINARIA	12-Jun-06	12-Jul-06	Miercoles 5-Jul-06 *
ETRAORDINARIA			Miercoles 23-Ago-06 *
ORDINARIA	28-Ago-06	14-Ago-06	Lunes 18-Sep-06 *
ORDINARIA	3-Oct-06	11-Sep-06	Lunes 27-Nov-06*

* Las Fechas de Sesion del Pleno Ordinarias y extraordinarias estan sujetas a confirmacion por lo cual se incluyen sólo de modo informativo

En esta opción podrá regresar al menú o salir del sistema.

Si eligiera la opción de consultar la documentación la pantalla le mostrará dicha información.



Consulta de Información de Estatus del Personal Académico y Periodos de Exámen por Semestre de la DIE

Documentación Para Realizar Algún Movimiento Académico

NUEVO INGRESO

- Acta de nacimineto original y dos copias.
- Tres copias del CURP.
- Tres copias del SAT.
- 6 Fotos a color tamaño infantil.
- 3 Copias de un comprobante de domicilio.
- 3 Formatos de curriculum (Proporcionados por el departamento).
- 3 Copias del Titulo (si es profesor de Asignatura).
- Carta Original de créditos y promedio con dos copias (Si es ayudante, con un 75% de creditos minimo y un promedio minimo de 8).

OTRO NOMBRAMIENTO

- 3 Copias de Título.
- Carta Original de creditos y promedio con dos copias.

PRORROGA

- Favor de notificar al departamento al que pertenece en el se encargaran de realizar la prorroga.

INICIO

SALIR DEL SISTEMA

En esta opción podrá regresar al estatus o salir del sistema.
Para salir del sistema existe en las pantallas la opción Salir.

Bibliografía

Bibliografía

Kendall, Kenneth & Julie, Kendall. "Análisis y Diseño de Sistemas", 3ª Edición, Ed. Prentice-Hall. México, 1991.

Senn, James. "Análisis y Diseño de Sistemas de Información",2ª Edición, Mc-Graw Hill, Abril 2000.

Braude, Eric J. "Ingeniería de Software: Una perspectiva orientada a objetos".Ed. AlfaOmega. México, 2003.

Pfleeger, Shari Lawrence. "Ingeniería de Software".Ed. Prentice-Hall. Buenos Aires, Argentina, 2002.

Pressman, Roger."Ingeniería de Software, Un enfoque práctico".4ª Edición, Ed. Mc-Graw Hill. España, 1998.

Viescas, John. "Running Microsoft Access 2000". Ed. Microsoft Press. U.S.A. 1999.

Horton, Ivor. "Beginning Java". Ed. Wrox Press, Canada.1997.

Koch, George. "Oracle: The Complete Reference". Ed. Mc-Graw Hill, U.S.A. 1991.